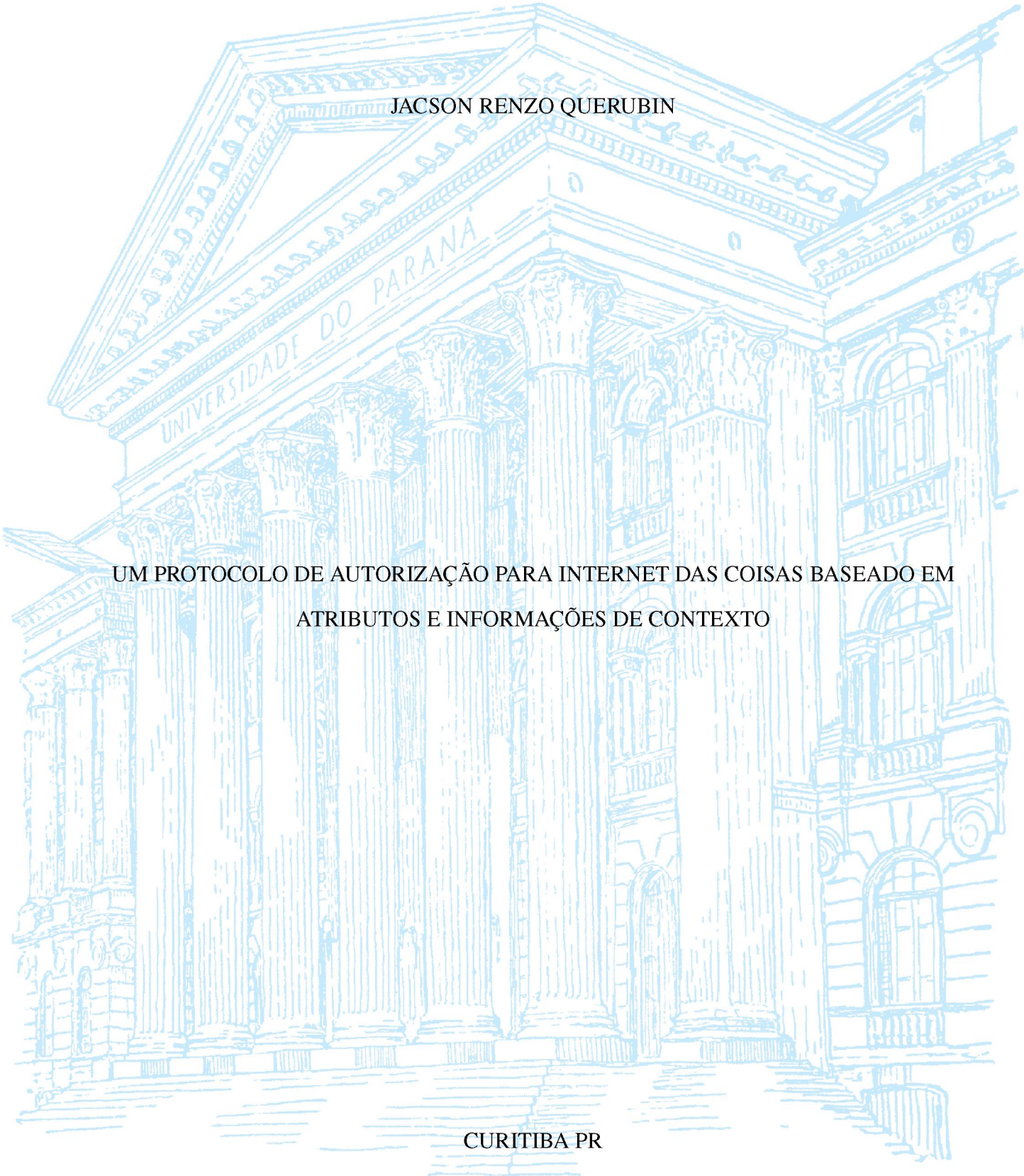
UNIVERSIDADE FEDERAL DO PARANÁ

JACSON RENZO QUERUBIN



UM PROTOCOLO DE AUTORIZAÇÃO PARA INTERNET DAS COISAS BASEADO EM ATRIBUTOS E INFORMAÇÕES DE CONTEXTO

CURITIBA PR

2019

JACSON RENZO QUERUBIN

UM PROTOCOLO DE AUTORIZAÇÃO PARA INTERNET DAS COISAS BASEADO EM ATRIBUTOS E INFORMAÇÕES DE CONTEXTO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação.*

Orientador: Carlos Alberto Maziero.

CURITIBA PR

2019

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

Q4p Querubin, Jacson Renzo
Um protocolo de autorização para Internet das coisas baseado em atributos e informações de contexto [recurso eletrônico] / Jacson Renzo Querubin – Curitiba, 2019.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática.
Orientador: Carlos Alberto Maziero.

1. Segurança - informática. 2. Protocolo de autorização - informática. 3. Controle de acesso (computadores). I. Universidade Federal do Paraná. II. Maziero, Carlos Alberto. III. Título.

CDD: 005.83

Bibliotecária: Roseny Rivelini Morciani CRB-9/1585

MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA - 40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **JACSON RENZO QUERUBIN** intitulada: **Um protocolo de autorização para Internet das Coisas baseado em atributos e informações de contexto**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 18 de Fevereiro de 2019.

CARLOS ALBERTO MAZIERO
Presidente da Banca Examinadora (UFPR)

ALTAIR OLIVO SANTIN
Avaliador Externo (PUC/PR)

ANDRÉ RICARDO ABED GRÉGIO
Avaliador Interno (UFPR)

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

A Karla, meu eterno amor.
A Peterson e Marlene, minhas inspirações.
A Elide (in memoriam) e Ataide (in memoriam), saudades.

”A man is not dead while his name is still spoken“
- Terry Pratchett, Going Postal, Chapter 4 prologue

AGRADECIMENTOS

Agradeço a minha esposa, Karla Noronha, pela inspiração diária e por me apoiar desde o princípio - *sou grato e muito sortudo de compartilhar essa existência contigo.* Agradeço a minha mãe, Marlene Querubin, pela eterno exemplo; ao meu irmão, Peterson Jardim, por ser quem é e proporcionar orgulho por isso; a Elide Cavinato e Ataide Querubin, por serem alicerces e eternos faróis; aos meus sogros, Anália e Pedro, que sempre apoiaram nas viagens a Curitiba. Extendo a toda família, pois cada um tem parte na minha história - *a todos e todas, meu muito obrigado.*

Agradeço ao orientador, Carlos Maziero, pela condução, paciência e disposição para me orientar (com a provável pequena dose de loucura, ao aceitar este aluno). Não há palavras que possam expressar a gratidão e aprendizado ao longo deste perído. Agradeço aos professores André Grégio e Altair Santin, membros da banca avaliadora, pelos comentários e sugestões de grande valia. Gostaria de agradecer o professor Luiz Carlos de Bona, pois, no intervalo de uma aula, conversando sobre o assunto, inspirou o tema deste trabalho. Além destes, gostaria de agradecer aos professores Fabiano Silva, Letícia Peres, Marcos Castilho e Roberto Pereira pela disposição de lecionar na Tríplice Fronteira. Extendo o agradecimento a todos os professores e professoras do PPGInf que contribuíram para realização deste MInter; aos colegas do LARSIS e do MInter pelos papos, amizade e apoio de sempre, meu obrigado. Agradeço a ITAIPU, UFPR e PTI/CELTAB, pela viabilização deste MInter. Meu eterno obrigado aos funcionários destas instituições pelo trabalho e dedicação.

Gostaria de agradecer a Marcos Dellazari, pelas inúmeras conversas e apoio; a Carlos Santiviago e Flávio Oliveira, pelos apoios e estudos em conjunto - foram muitos litros de café, horas estudando e trocando idéias. Gostaria de agradecer a Igor Mussoi e Washington Medeiros que prontamente aprovaram os anseios deste eterno aluno. Gostaria de registrar o meu muito obrigado à equipe da SIPS, que deram apoio nos momentos que estávamos dedicados aos estudos. Não poderia deixar de agradecer a Marcos Siríaco pelo apoio e ajuda, sempre com um sensor ou componente para emprestar nos horários mais absurdos. Pela minha disponibilidade durante o início deste Mestrado, reforço meu agradecimento a ITAIPU.

Agradeço os amigos e amigas, sem nomear, pois, tenho a felicidade de ter tantas pessoas queridas. Feliz é a pessoa que tem amigos. *Obrigado!*

Registro aqui meu agradecimentos a Hermann (*in memoriam*), Bilú (*in memoriam*), Sartre (*in memoriam*), Eddie, Freud, Janis, Lola, Billy, Zé, Sortudo, Edmunda, Moby, Aston, Catarina e Apollo - a estes e todos os serezinhos que nos ensinam neste curso da vida, *obrigado.*

Por fim, gostaria de lembrar que há mais pessoas que ajudaram este trabalho do que palavras aqui presentes. Esta obra é fruto de uma série de pessoas e eventos que me auxiliaram diretamente e indiretamente. Por chegar até aqui, agradeço a todos os professores e professoras de minha vida, sejam estes de profissão ou não.

PS: Agradeço você que está lendo essa dissertação, obrigado pela atenção.

RESUMO

Com a Internet das Coisas, a ubiquidade de dispositivos cresce a cada dia. Controlar e garantir comunicação segura e amigável ao usuário final continua um desafio. Este trabalho propõe um protocolo de autorização para dispositivos baseado em atributos e informações de contexto. As informações de contexto são observações disponíveis em sensores dos dispositivos, tais como: luz, temperatura, microfone - para citar algumas. Este contexto provê elementos que auxiliam nas decisões de segurança, como fatores complementares de autorização. Agrega-se ao contexto um fator social dos dispositivos, auxiliando na avaliação de acessos e respectivos riscos. Este fator social consiste tanto no histórico das observações realizadas pelo dispositivo, quanto no cumprimento de suas funções (comportamento esperado). De posse destas informações de contexto e de atributos do pedido (requisitante e risco da informação), o protocolo classifica o risco e decide se efetiva ou não o pedido. As informações de contexto são base para o modelo de confiança entre os dispositivos, sendo utilizadas como validações entre os mesmos. Somado ao protocolo apresentado, discute-se o modelo de ameaças e respectivas mitigações as quais o protocolo resiste satisfatoriamente. Na sequencia, apresenta-se o uso e análise específica de um atributo: a luminância. Discute-se o uso da luminância para representação de um mesmo ambiente e respectivos resultados desta pesquisa. Por fim, algumas considerações e possibilidades de pesquisa em aberto.
Palavras-chave: IoT, Segurança, Autorização, ABAC, Contexto, Protocolo, Luminância

ABSTRACT

Internet of Things grows the devices' ubiquity constinuously. It remains a challenge to control and ensure a user-friendly secure communication. This work presents an context-aware attribute-based authorization protocol. The context provides elements that helps decision making. The devices' social factor are included in the context, assisting in access evaluation and its risks. Context information is gathered throught devices' sensors, such as: light, temperature, sound - to list some examples. This context provides aid in security decisions, as complementary factors of authorization. A devices' social factor is added to the context, aiding the accesses evaluation and its risks. This social factor consists both the observations historical made by devices, and in by execution of its functions (expected behavior). With this context information and request attributes (asker and information risk), the protocol classifies the risk and decides whether or not will honor the request. Context information is the basis for the trust model between devices, and is used as validation between devices. In addition to the presented protocol, the threat model and its mitigations are discussed, which the protocol resists satisfactorily. In sequence, the analysis of an attribute use is presented: luminance. We discuss the luminance's use to represent same environment. Finally, we conclude with some considerations and open research possibilities.

Keywords: IoT, Security, Authorization, ABAC, Context-Aware, Protocol, Luminance

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

ABAC	Attribute Based Access Control
ACL	Access Control Lists
AmI	Ambient Intelligence
AMQP	Advanced Message Queuing Protocol
AoT	Android of Things
AS	Authentication Server
BSM	Broadband Satellite Multimedia
BUG	the Good, the Bad and the Ugly
BT	Blink Timetout
CA	Certificate Authority
CapBAC	Capability Based Access Control
CDMA	Code Division Multiple Access
CoCapBAC	Community Capability-Based Access Control
CXaaS	Context-as-a-Service
DAC	Discretionary Access Control
DNS	Domain Name Service
DNSSEC	DNS Security Extensions
DVB-S	Digital Video Broadcasting-Satellite
EDR	Enhanced Data Rate
FDMA	Frequency Division Multiple Access
FTBAC	Fuzzy Trusted Based Access Control
GPRS	General Packet Radio Service
GPS	Global Positioning System
HSPA+	Evolved High Speed Packet Access
IdM	Identity Management
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
KMS	Key Management Systems
LED	Light-Emitting Diode
LT	Long Timeout
LTE	Long Term Evolution
M2M	Machine-to-Machine

MAC	Mandatory Access Control
MITM	Man-in-the-Middle
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MS	Mafia Score
NFC	Near Field Communications
PDP	Policy Desicion Point
PEP	Policy Enforcement Point
PKI	Publick Key Infrasctructure
PoC	Proof Of Concept
POSIX	Portable Operating System Interface for Unix
PoW	Proof-of-Work
QoS	Quality of Service
QRCODE	Quick Response Code
RAdAC	Risk-Adaptable Access Control
RBAC	Rule Based Access Control
ReBAC	Relationship-Bases Access Control
REST	Representational state transfer
RFID	Radio Frequency Identification
RGB	Red-Green-Blue
RPi	Raspberry Pi
RSN	RFID Sensor Network
SDN	Software Defined Network
SIGINT	SIGnals INTelligence
SOA	Service Oriented Architecture
SPOF	Single Point of Failure
SSP	Secure Simple Pairing
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
TOR	The Onion Router
UCON	Usage CONtrol
UTMS	Universal Mobile Telecommunications System
WiMAX	Worldwide Interoperability for Microwave Access
WoT	Web of Things
WSN	Wireless Sensor Networks
WSAN	Wireless Sensor and Actuator Networks
XMPP	Extensible Messaging and Presence Protocol
xDSL	“any” Digital Subscriber Line

SUMÁRIO

1 INTRODUÇÃO

Na última década, a computação cada vez mais se aproximou do cotidiano das pessoas através de interações nos mais variados ambientes - incluindo o corpo humano. No início, meados da década de 40, a computação era composta de dispositivos extremamente grandes e de pouca interatividade; entretanto, com o passar dos anos, as iterações evolutivas da computação se aproximam e interagem cada vez mais com as pessoas e os ambientes, das mais variadas formas. Conforme observado na Figura 1.1, vê-se que a cada iteração aumentaram-se a complexidade e os inúmeros requisitos necessários para satisfação das necessidades e anseios das pessoas, através de respostas rápidas nos mais variados contextos dinâmicos do conjunto de: pessoas, ambientes, processos e tecnologias [1, 2, 3, 4, 5].

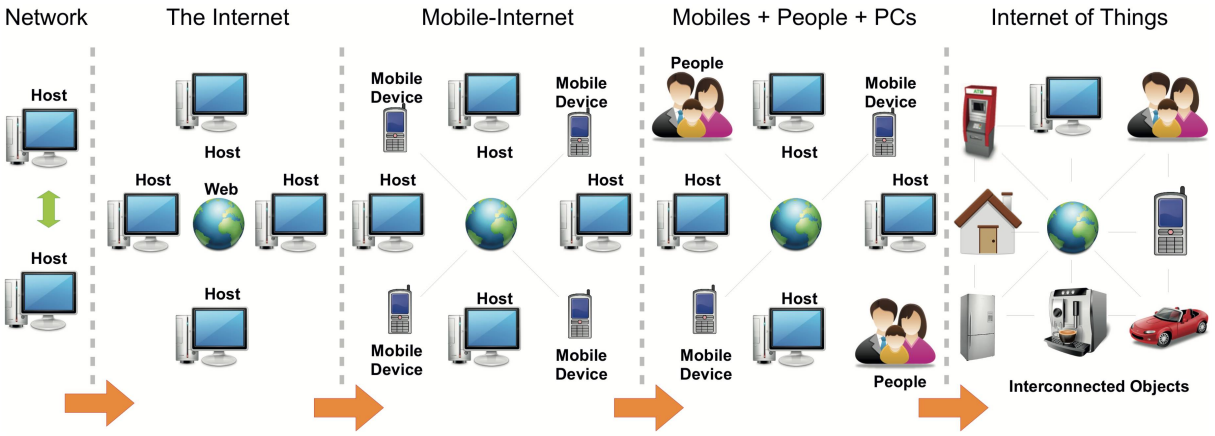


Figura 1.1: Evolução da Internet em 5 fases [5].

A "Internet das Coisas" caracteriza a presença pervasiva de computação em nossa volta - tratada de maneira genéria como ***coisas*** *(things)*, neste texto estas serão referenciadas também como **objetos** (*smart objects*) ou **dispositivos**. Do ponto de vista lógico, o termo Internet das Coisas (referenciado aqui como `IoT` - *Internet of Things*) consiste no paradigma de comunicação entre dispositivos inteligentes que interagem colaborativamente para chegarem a um objetivo comum. Do ponto de vista tecnológico, IoT é a conectividade global entre todas estas entidades heterogêneas únicas - incluindo todo aparato tecnológico que suporta estas. E por último, o ponto de vista comercial, grande motor deste novo paradigma, que utiliza-se dessa conectividade e dispositivos para prestação de novos tipos de serviços e, consequentemente, novas oportunidades de negócios [6, 7, 8, 9].

Por mais estranho que pareça, o uso de *coisas* como terminologia, se deve ao fato de que os dispositivos que compõem a IoT são caracterizados por ambas capacidades reduzidas: computacional e autonomia energética. Torna-se um desafio transformar esta tecnologia em algo *invisível*, para uso das pessoas. Para atingir este objetivo, os dispositivos devem: entender a situação dos usuários e em quais ambientes se encontram, comunicar-se com plataformas ou compartilhar informações entre si e realizar as devidas ações conforme o contexto apresentado. Tecnicamente, para conseguir atingir estas características, os objetos devem não apenas monitorar (*sensing*) e interagir com o ambiente (*actuating*), mas usar os padrões de internet já estabelecidos para prover serviços de transferência, análise e a sintetização dos dados coletados, ou até mesmo criados, geralmente em redes heterogêneas e esparsas. A complexidade se faz inerente, já que os

dispositivos têm diferentes características operacionais, tais como taxas de amostragem, taxas de erros e tipos de sensores [10, 9, 11].

Conforme existam esta geração de dados, comunicação e sintetização de informações, o choque da IoT na vida das pessoas é enorme. Um dos pontos que merece atenção é a segurança. Os conceitos e técnicas que a área de segurança pesquisou agora precisam ser avaliados em sua aplicabilidade na IoT, acarretando em uma nova análise dos novos desafios e novos requisitos. A restrição de poder computacional e uso energético pelos objetos fazem com que soluções que não façam uso eficiente destes pontos sejam descartadas - ou refeitas para se adequar a nova realidade de uso.

A IoT tem relação direta com privacidade, o grande desafio deste paradigma. Visões distintas das mais variadas entidades envolvidas provocam novos conflitos e efeitos. O impacto de decisões sobre privacidade nas propostas de implantação de IoT, tem influência direta na aceitação das pessoas, e consequentemente nas oportunidades de negócios envolvidas [12, 8].

1.1 MOTIVAÇÃO

Conforme apresentado, capacidade limitada dos dispositivos induz a uma análise mais precisa dos métodos e propostas de autorização e controle de acesso. A aplicação destes em um ambiente, tais como: uma casa, linha de produção, bairro, universidade, entre outros - consistem em uma avaliação pertinente. O uso destes dispositivos no dia-a-dia das pessoas provoca um conflito entre duas premissas: praticidade e privacidade. Por exemplo, para termos uma praticidade de uso, os sistemas oferecem ajuda dependendo do contexto em que a pessoa/dispositivo se encontram (analisando por exemplo o ponto geográfico e hora do dia). Neste exemplo já temos um problema de privacidade, pois ao ser analisado ou estudado esse comportamento, gera um conjunto de registros das localidades e hábitos de uso - tornando um possível ponto de vazamento de dados privativos (*profiling*) [8, 13, 14, 6].

Quanto mais sensível é o ambiente onde estes dispositivos operam, maior se torna o impacto do vazamento dos dados observados. A privacidade de operação em casa (ambiente íntimo) ou em um hospital, consistem em dados altamente críticos, no que tange à privacidade. Os casos em que há a quebra da "confiança" de uso com privacidade, influenciam diretamente nos negócios, às vezes com impactos jurídicos. Devido à sensibilidade destes dados, incrementa-se a motivação de espionagem (*eavesdropping*) - executada por agentes de estado ou particulares.

Um caso recente demonstra bem esse oceano de dados pode ter consequências não previstas: um aplicativo de acompanhamento de exercícios publicou dados "anonimizados" dos exercícios praticados pelos usuários. Olhando no mapa em modo de imagem de satélite, Jack Nelson[1] começou a correlacionar trajetos em locais insólitos no globo. Assim conseguiu identificar bases militares de vários países. Outros, correlacionando outras informações e cruzando dados, conseguiram identificações de pessoas, incluindo informações críticas, como os batimentos cardíacos destas [15, 16, 17].

Criar um ambiente prático para usuário final e ao mesmo tempo seja seguro por padrão, é um desafio que ainda persiste. Pesquisadores buscam solucionar este desafio, de forma que se ofereça ao usuário a possibilidade de usar dispositivos em seus mais variados ambientes, com praticidade e tranquilidade de estar seguros.

[1] https://bit.ly/2GFvuWI

1.2 PROBLEMA

Essa dicotomia entre praticidade *versus* privacidade é tema constante de pesquisas, tendo impacto direto na satisfação e expectativa dos usuários. Conforme já posto, a segurança tem um impacto na adoção tecnológica e criam-se novos desafios. Um ponto importante, durante o ciclo de uso de um dispositivo, consiste no gerenciamento de identidade (*Identity Management* - IdM). O IdM engloba o gerenciamento, autenticação, autorização e respectivos privilégios, limitados as fronteiras dos sistemas de um mesmo domínio [18, 19].

Em um domínio de segurança, estas etapas do ciclo de vida de uma identidade compõem um processo que relaciona pedidos e acessos. Para autorizar uma identidade (seja uma pessoa ou dispositivo) acessar um objeto (tendo portanto, um privilégio), são necessários mecanismos de controle, a autorização. Existem vários modelos propostos, alguns mais centralizados, outros distribuídos - incluindo federados. Independente do modelo de autorização, ainda persiste a dificuldade de uso dos usuários comuns, bem como configurações inseguras de fábrica [14, 9, 10, 18].

Algumas pesquisas estão focando em mecanismos que facilitem a autorização através de informações complementares, que auxiliariam na classificação do risco do pedido de acesso de um dispositivo. Este mecanismo pode utilizar áudio, imagem, entre outras informações do ambiente. Se faz uma possibilidade interessante pesquisar se estas informações de fato auxiliam tanto na segurança, quanto na praticidade de uso [20, 21].

1.3 HIPÓTESE

Conforme apresentados, alguns fatores do ambiente em que dispositivos de IoT se encontram (áudio, câmera, luminosidade, temperatura, interações "sociais", geolocalização, entre outros) proveem um contexto de uso e propriedade. Assim surgiu a possibilidade de pesquisar a viabilidade de uso destas informações de forma complementar na etapa de autorização, do gerenciamento de identidade do(s) ambiente(s). Com isso, tem-se a seguinte hipótese de pesquisa:

> Qual a viabilidade do uso das informações de contexto para prover maior segurança na etapa de autorização?

Outras dúvidas desdobram-se desta hipótese: caso seja possível usar estas informações, facilita-se o uso? Quantas leituras e quais tipos de observações? Conforme estas questões, são apresentados alguns objetivos a seguir.

1.4 OBJETIVOS

O objetivo principal deste trabalho é avaliar as informações de contexto e o uso dos respectivos fatores via um protocolo de autorização para dispositivos IoT. Além deste, outro objetivo consiste no suporte as informações: luminância do ambiente, temperatura, som e câmera - além de extensível a outras classes de informações, também. Somado ao conjunto, agrega-se suporte ao contexto um componente "social". Este componente social é a combinação da confiança e análise de comportamento dos dispositivos do domínio. Desta forma os dispositivos - baseados em informações do ambiente, grupo e atributos do pedido - analisam os pedidos de autorização e efetua-se a decisão conforme a política definida. Em posse deste conjunto de informações - atributos, ambiente, fatores sociais e pedido, autoriza-se ou revoga-se o acesso a algum objeto e/ou recurso.

Outro objetivo consiste em avaliar o modelo de ameaças do protocolo e respectivas mitigações do mesmo. Por último, disponibilizar um código base disponível para outras propostas e avaliações, devido à dificuldade de código específico para esse fim estar disponível para comparação.

1.5 ESTRUTURAÇÃO

Este trabalho tem a seguinte estrutura: o capítulo 2 apresenta a fundamentação teórica com um panorama sobre IoT: usos, desafios, contexto, requisitos de segurança e os trabalhos correlatos; na sequência, o capítulo 3 apresenta a proposta deste trabalho com a arquitetura e escopo. No capítulo 4, apresenta-se o modelo de confiança, a gestão de confiança, risco entre dispositivos, ciclo de vida de dispositivos e o ciclo de vida do domínio. O capítulo 5, são apresentados os detalhes de implementação, avaliação da segurança, o modelo de ameaças e o uso dos atributos de contexto. Por fim, o capítulo 6 conclui-se com uma discussão sobre os resultados, trabalhos futuros e desafios em aberto.

2 FUNDAMENTAÇÃO

Este capítulo apresenta o embasamento teórico para a concepção e uso do protocolo. Primeiro fundamenta-se uma visão geral da Internet das Coisas, com conceitos iniciais, usos, modelos abordados e principais aplicações - além das possibilidades novas de negócios. Após isto, fundamenta-se os conceitos de contexto na computação e respectivas especificidades da IoT. Por último, fundamenta-se os conceitos de segurança aplicados à IoT, destacando-se o controle de acesso - área que este trabalho participa.

2.1 INTERNET DAS COISAS

Conforme visto no capítulo 1, a *Internet das Coisas* caracteriza a presença pervasiva de computação em nossa volta. Outra definição do termo tem o sentido de que há uma interconexão mundial entre objetos heterogêneos, com endereçamento único, usando protocolos padronizados. Porém, cabe frisar, que IoT implica em uma visão mais abrangente do que a ideia de somente identificar objetos. A IoT é construída sobre três pilares: ser identificável, ser comunicável e ser interativa [6, 9]. Define-se estes objetos, ou *coisas*, como entidades que [9]:

- Mantêm uma delimitação física;
- Mantêm uma capacidade mínima de comunicação (descoberta e interação);
- Contêm um identificador único;
- São associadas com, pelo menos, um nome e um endereço;
- Possuem capacidade computacional básica;
- Podem monitorar um fenômeno físico.

Outro termo relacionado aos objetos é o *spime* (*space* + *time*, tempo e espaço fundidos), definido como um objeto que pode ser rastreado através da sua localização e tempo, durante toda o sua vida de maneira sustentável, aprimorável e unicamente identificável [6].

Por outro lado, há a definição de ambientes inteligentes (ou *smart environments*). Estes consistem na interconexão de dispositivos e sensores que habilitam o compartilhamento de informações de plataformas através de uma estrutura unificada, desenvolvendo uma visão comum para aplicações inovadoras. Do ponto de vista sistemático, a IoT pode ser vista como um sistema altamente dinâmico e radicalmente distribuído, composto de um número gigantesco de objetos produzindo e consumindo informação. Do ponto de vista das pessoas, IoT irá habilitar uma grande quantidade de serviços *sempre responsivos*, os quais devem responder às suas necessidades e apoiá-las em suas atividades diárias [10, 9]. De forma resumida, IoT deve, como características chaves, suportar:

- Heterogeneidade de dispositivos;
- Escalabilidade;
- Troca de informações pervasiva via tecnologias sem-fio e de proximidade;
- Eficiência energética;

- Capacidade auto-organizativa;
- Gerenciamento de informações e interoperabilidade semântica;
- Segurança embarcada e mecanismos de preservação de privacidade.

Resumindo, na IoT qualquer objeto terá três características: este se comunica, se identifica e interage com o meio (e outros objetos). Ultimamente, o maior desafio na perspectiva computacional e comunicativa se coloca na necessidade de que a arquitetura proposta deve suportar um consumo baixo de energia, um custo baixo e, ainda assim, ser totalmente conectada e integrada - regida através de padrões já existentes [9].

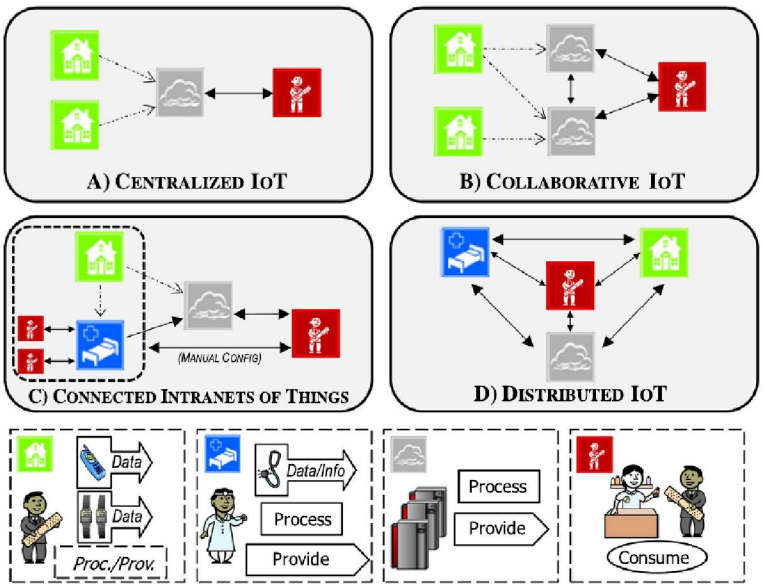


Figura 2.1: Visão geral dos modelos de IoT [22].

Conforme a Figura 2.1, existem algumas abordagens distintas para uma arquitetura de IoT [22]:

A) **IoT Centralizada**: as coisas são passivas e o acesso aos dados é realizado através de uma interface central (geralmente em nuvem);

B) **IoT Colaborativa**: as coisas são passivas e o acesso aos dados é realizado através de serviços que colaboram entre si (também em nuvem);

C) **IoT Conectada**: as coisas são ativas e processam informações no ambiente, enviando estas para provedores centrais e usuários (locais ou remotos);

D) **IoT Distribuída**: todas as entidades envolvidas (coisas, usuários e serviços) capturam, processam e sintetizam informações, provendo estas a outras entidades.

Cada modelo oferece vantagens e desvantagens, porém o item D é o que mais se aproxima da realidade e uso, incluindo os desafios (complexidade e interoperabilidade). Para sua aplicação, a IoT é compostas por alguns elementos, que apresenta-se mais a fundo, na próxima seção.

2.1.1 Elementos da IoT

Existem três camadas de IoT que habilitam a computação ubíqua em si: *hardware*, *middleware/framework* e apresentação/aplicação. A primeira camada, *hardware*, consiste nas tecnologias responsáveis pela interação com o usuário ou a "leitura" do ambiente. São responsáveis também pela atuação - quando possível - e realizam a comunicação entre os dispositivos de forma eficiente. A Figura 2.2 mostra a relação entre IoT e as redes de sensores. As tecnologias mais encontradas no componente de *hardware* são [6, 23, 24, 10, 5]:

- Tecnologias para coletar dados:
  - *Radio Frequency Identification* (RFID) e *RFID Sensor Network* (RSN): até 640 kbps, entre 3-10m de distância;
  - *Wireless Sensor Networks*(WSN) e *Wireless Sensor and Actuator Networks* (WSAN): até 250 kbps, entre 10-100m;
  - *Bluetooth*: 1-24 Mbps,
  - *Near-Field Communications* (NFC): 106-424 kbps, menor que 10m de distrância.
- Tecnologias de acesso:
  - *Ethernet* - 802.3 (u/z);
  - *Wifi* - 802.11 (a/b/g/n);
  - *WiMAX* - 802.16 (a/d/e/m);
  - *xDSLs* e *Smart Gateway*;
  - *Acesso*: CDMA,TDMA e FDMA;
  - *Cellular* - GSM, GPRS, UTMS, HSPA+, LTE;
  - *Satelite* - BSM, DVB-S; e
  - *PLC* - DVB-TS, HomePlug AV, IEEE 1901;

Um desafio na camada de *hardware*/comunicação consiste no uso do TCP. O TCP contém características que tornam ineficiente o seu uso em IoT, tais como a configuração da sessão, controle de congestionamento e uso de memória (*buffer*)[6].

O componente de *middleware* realiza a abstração da representação da heterogeneidade de objetos e suas características (capacidades, formas de comunicação, entre outros) para o programador - facilitando o desenvolvimento de novas aplicações. Entre os objetivos do *middleware*, destacam-se a abstração de objetos, gerenciamento de serviços e a composição destes. Esta composição é conhecida como *Service Oriented Architeture* (SOA), com seus pilares que permitem decompor serviços complexos em componentes menores com interfaces bem definidas.

Outra vantagem dos *middlewares* consiste na capacidade de suportar novos *hardwares* e serviços de forma mais rápida - provendo uma flexibilidade interessante para alterações no mercado. Uma classe específica de *middlewares* consiste nos *frameworks*. Estes consistem em uma versão mais "enxuta" dos primeiros, pois não oferecem serviços e são somente uma facilidade de programação para o desenvolvedor [6, 25, 26].

No caso da camada de apresentação, esta consiste em formas inovadoras de visualização dos dados coletados e tratados pelo *middleware*. E mais, nesta consistem as ferramentas de interpretação dos dados - as quais serão acessadas através das mais variadas formas e dispositivos.

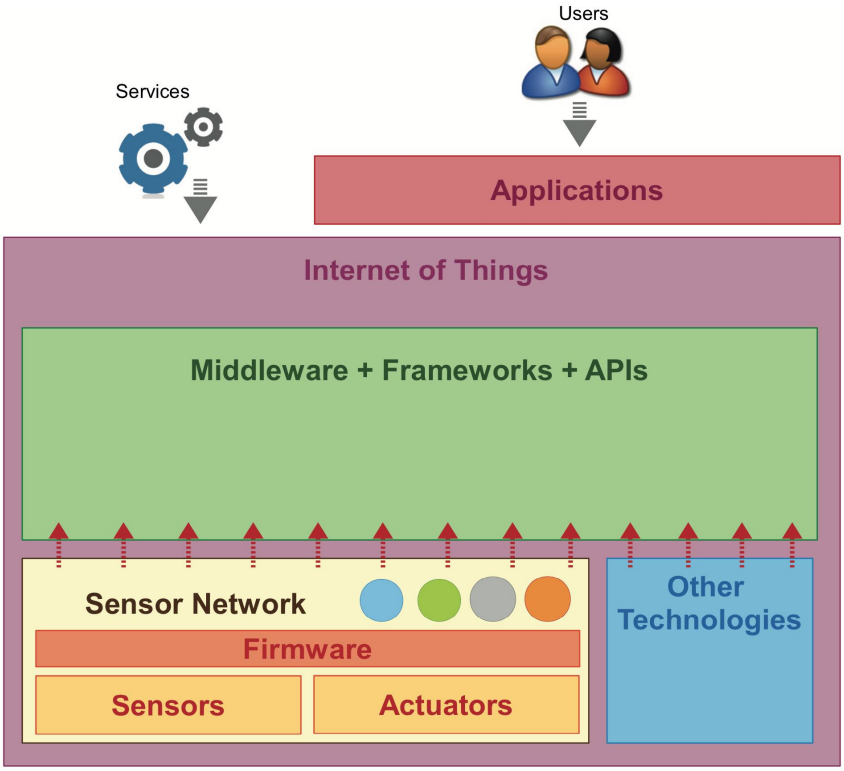


Figura 2.2: Relação entre IoT e redes de sensores [5].

Geralmente, as duas camadas mais superiores (*middleware* e apresentação) são aglutinadas em uma plataforma [10].

Seja como *middleware* ou plataforma, existem desafios que precisam ser resolvidos pelos mesmos, tais como: interoperabilidade, escalabilidade, abstração, interação entre objetos, infraestrutura flexível, multiplicidade e segurança/privacidade. Em alguns cenários agregam-se ainda outros desafios, como: imenso número de dispositivos e eventos, interações espontâneas[1], análise de contexto[2], análise espacial, recursos limitados e QoS. Atualmente tem-se uma miríade de propostas de plataformas e *middlewares* - um campo em constante efervescência - destacam-se alguns exemplos como: *Android of Things* (AoT) [27], *Web of Things* (WoT) [28], HYDRA[29], UBISOAP/UBIROAD[25], ASPIRE [25]. Estes são alguns poucos exemplos, atualmente existem dezenas de propostas de *framework* e plataformas [30, 29, 25].

Alguns trabalhos apresentam um bom comparativo entre as propostas de plataformas/*middlewares*, porém há desafios ainda em aberto: banco de dados relacionais *versus* respostas em tempo real; mecanismos de composição pré-definidos e determinísticos que não escalam de forma eficiente em redes com um número enorme e muito dinâmico de dispositivos, são alguns exemplos. Cada proposta foca em resolver um subconjunto ou sacrifica-se outros, não há uma "bala de prata" que resolva todos os desafios até o presente momento [30, 29, 24].

Outro ponto ainda sem solução *de facto*, consiste nos simuladores. Já existe a demanda de simuladores de IoT para acelerar o desenvolvimento e testes das propostas de *middleware*/plataformas. Alguns requisitos básicos já foram analisados e apresentados, tais como: identificador único (ou *tokens*) para dispositivos, geração de dados e metadados de sensores (em

[1]Aplicações IoT tem a característica de que, em um dado momento, seja por movimentação ou reorganização, os objetos entram ao alcance de outros - na rede *mesh* - ativando eventos sem tempo ou "compasso" esperado.

[2]Os objetos podem gerar dados conforme ambiente ou o processamento dos dados coletados muda conforme ambiente, hora da coleta, entre outros parâmetros.

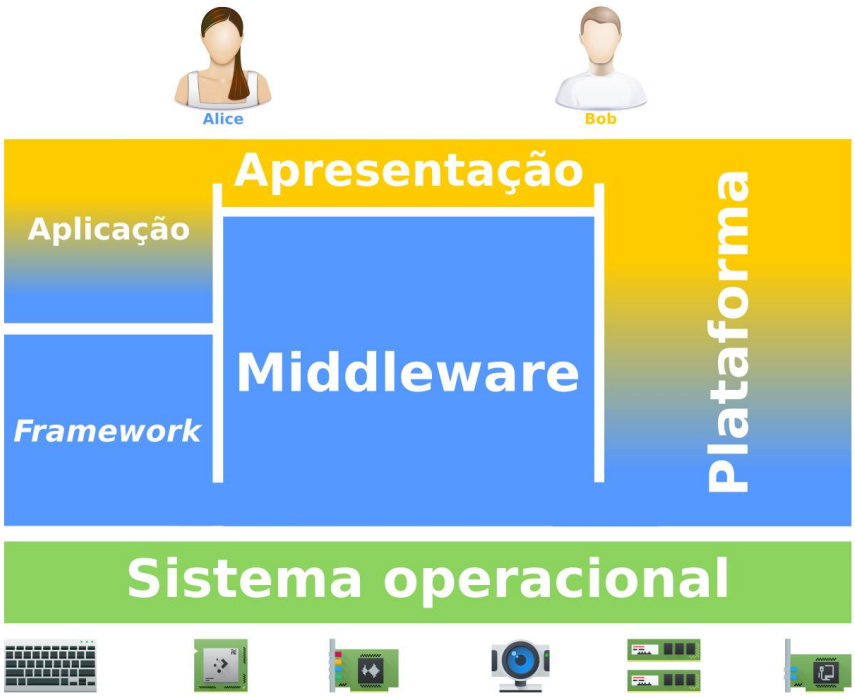


Figura 2.3: Elementos da IoT

formatos binários, texto ou JSON) e protocolos de comunicação (MQTT ou REST). O *IOTSim* (baseado no *Cloudsim*) é uma das únicas propostas até o momento [31, 32].

O contexto é um ponto crucial para geração e tratamento dos dados e seus metadados - influenciando diretamente na comunicação entre objetos. Uma ampla visão apresentada enumera as características e principais desafios com os dados da IoT: aquisição, modelagem, análise e distribuição [5].

Outras tecnologias ganharam força com a maior utilização da IoT. Devido à enorme geração de dados, estas são necessárias para processar e extrair informações relevantes, tendências e previsões. Uma destas tecnologias é o *Big Data*. Devido ao enorme volume, algumas previsões apontam 35 ZB para 2020, o *Big Data* é uma das tecnologias fundamentais para distinguir entre dados valiosos de ruídos. Porém desafios aparecem com o seu uso: alto uso de energia para processamento e respectiva refrigeração, análise adaptativa de dados em tempo real, projetar um armazenamento escalável e eficiente, garantir segurança dos dados coletados e a privacidade, resumir/sintetizar os dados coletados, otimizar pesquisas e prover multi-*tenants* [33, 29, 30].

Já o *Machine Learning* (ML) consiste em aprender uma tarefa particular. Para aprender a tarefa o ML utiliza um conjunto de exemplos para relacionar a entrada com a saída esperada. Assim, enquanto tarefas específicas, modelos e a categorização mudam entre as aplicações, geralmente estas recaem sob as seguintes categorias: classificação, regressão e detecção de anomalias. Um exemplo de uso seria métodos de ML que descobrem correlações importantes nos dados de sensores e propõem uma disposição melhorada para máxima cobertura dos mesmos [34, 35].

Por sua natureza distribuída, a IoT enfrenta vários desafios, pois tem-se problemas de eleição de líder, de contagem de nós e de votação de leituras, problemas centrais da literatura para superar falhas em *hardware*, tais como: leitores e coletores defeituosos; ou até uma média das leituras. Inclusive algoritmos de escalonamento e alocação dinâmica de recursos baseados no princípio de enxames otimizados estão sendo desenvolvidos (uma forma de aplicar um "conhecimento da natureza" para solução de problemas técnicos). Essa otimização se faz

necessária, já que problemas de escalabilidade desencorajam o uso de uma comunicação com vários saltos (*Multi-hop*) para recuperar as leituras. Já o controle da informação que os objetos produzem e enviam, em conjunto com técnicas de filtragem, é uma preocupação em cenários pervasivos. As tecnologias citadas até aqui são os principais elementos e ferramentas para o uso e adoção da IoT. A seguir, são elencadas algumas aplicações da IoT nos mais variados contextos [9, 10].

2.1.2 Aplicações

O uso de IoT se dá cada vez mais pervasivo em nossa volta, com alguns usos já em escala, e outros com demandas crescentes. A seguir exemplos das principais aplicações e usos, bem como suas interações [36, 6, 37]:

- **Saúde**: hospitais inteligentes, casas inteligentes para doentes, saúde "conectada", identificação e autenticação de pessoal e pacientes, análise de dietas de forma dinâmica, previsão e prevenção de doenças;
- **Bem-Estar**: casas ou escritórios confortáveis, academia de ginástica inteligente, interações sociais, objetos perdidos, furtos, ambiente de jogos dinâmico;
- **Educação**: salas digitais, registro de frequências;
- **Turismo**: informações turísticas, museus e galerias inteligentes, gerenciamento de quartos/hóspedes inteligente;
- **Agricultura**: semeadura e colheita, diagnósticos, manutenção, produção, análise distribuída de demanda;
- **Financeiro**: gerência de risco (baseada em hábitos), detecção de fraudes, micropagamentos e microtransações para serviços dinâmicos;
- **Manufatura**: controle de estoque inteligentes, produção eficiente, análise dinâmica de demanda;
- **Transporte**: logística, direção segura, manutenção veicular, estradas ou ferrovias inteligentes, monitoramento de contêineres e pacotes, carros autônomos, pontos inteligentes, mapas em realidade aumentada;
- **Energia**: medidores residenciais inteligentes, leitores de oleodutos inteligentes, iluminação municipal inteligente, M2M para leilões de consumo e produção de energia (residencial/comercial/industrial);
- **Governo**: cidades inteligentes, governança eficiente, gerenciamento de tráfego (urbano e estradas).

2.2 CONTEXTO NA IOT

O conceito de contexto ubíquo na computação tem duas origens. Por um lado, é uma noção técnica, que oferece aos desenvolvedores novas formas de conceituar a ação humana e a relação entre esta ação e os sistemas computacionais que a suporta. Por outro, é, ao mesmo tempo, uma noção vinda das ciências sociais, trazendo uma atenção analítica para certos aspectos

sociais. Traduzir ideias entre diferentes domínios intelectuais pode ser tanto excepcionalmente valioso quanto inesperadamente difícil [38].

Alguns trabalhos classificam contexto através dos sinônimos ambiente e situação. Porém estas não auxiliam com precisão a definições de um novo "contexto". Se faz necessária uma definição mais precisa desse conhecimento do contexto, que alguns classificam como "consciente". Aqui empregar-se-á a definição conforme [5]:

Contexto: consiste em qualquer informação que possa ser usada para caracterizar a situação de uma entidade;

Entidade: consiste em uma pessoa, local ou objeto considerado relevante para a interatividade entre usuário e aplicação, incluindo estes;

Sistema baseado em contexto: um sistema é considerado baseado em contexto quando utiliza o próprio contexto para prover informações relevantes e/ou serviços para o usuário, sendo que a relevância depende da tarefa que o usuário está realizando;

Modelo de contexto: um modelo de contexto identifica um subconjunto do contexto que é realisticamente alcançável a partir de sensores, aplicações e usuários. Sendo, assim, possível de ser usado para execução da tarefa;

Atributo de contexto: é um elemento do modelo de contexto que descreve o mesmo. Um atributo de contexto tem um identificador, um tipo e um valor; opcionalmente contém uma coleção de propriedades descrevendo características específicas.

Assim, o dispositivo pode usar algumas categorias de contexto para analisar o ambiente que se encontra. Categorias como localização, horário, atividade e identidade são usadas direta ou indiretamente. Por exemplo, um dispositivo pode utilizar sua localização via GPS para saber onde se encontra (contexto primário). Caso este utilize duas leituras de pontos do GPS em tempos distintos (combinando categorias, portanto), poderá calcular seu próprio deslocamento (e saber se está em movimento, por exemplo). No caso, a distância se caracteriza como contexto secundário, pois foi "derivada" do contexto primário (inclusive, pode-se requisitar um serviço de mapas para conseguir renderizar o mapa e o caminho percorrido) [5, 39, 40].

Outro exemplo é o microfone. Ao comparar o áudio capturado de dispositivos distintos, valida-se o caso da simultaneidade de ambiente ou não. Observa-se que os sensores auxiliam os dispositivos na aquisição mais detalhada de informações do ambiente (e, portanto, contexto). Modeladas, estas informações podem ser analisadas, adequando o comportamento do dispositivo. Caso haja uma coordenação entre dispositivos para disseminar este contexto, fecha-se o ciclo de vida do mesmo [5].

2.2.1 Ciclo de vida do Contexto

Um ciclo de vida de dados demonstra como estes dados se movem entre as fases de um sistema. Este ciclo explica onde o dado é gerado e quando o mesmo é consumido. O movimento entre contextos não se encontra limitado aos *desktops*, *web* ou aplicações móveis - já existe, inclusive, Contexto como Serviço (CXaaS). Observa-se que o gerenciamento de contexto se tornou uma funcionalidade essencial para sistemas - tendência que deve aumentar com a IoT. O ciclo de vida de um contexto consistem em 4 partes: aquisição, modelagem, análise e disseminação.

A Figura 2.4 apresenta visualmente as partes do ciclo de vida de contexto. A aquisição é realizada através de sensores físicos ou virtuais. A seguir, os dados devem ser modelados

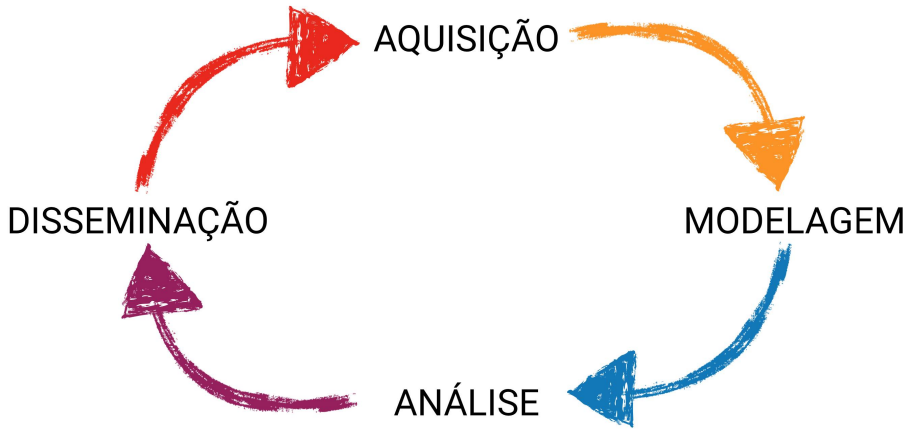


Figura 2.4: Ciclo de vida de um contexto [5].

e representados de forma adequada e proposital. Depois de modelados, os dados devem ser processados para transformar informações cruas de sensores em conhecimento de contexto, ou seja, uma visão em alto-nível. Por último, tanto esta visão em alto-nível do contexto, quanto os dados mais crus, devem ser disseminados aos envolvidos neste contexto [5, 40].

Outro conceito relacionado ao contexto é a Inteligência do Ambiente (AmI[3]). A AmI é um conceito o qual o ambiente digital (rede de dispositivos e sensores) interage, ajuda, "sente", auxiliando as pessoas no dia-a-dia. AmI é invisível, inteligente e flexível. Para atingir estes objetivos, AmI é composta por computação ubíqua e pervasiva, sensores, comunicação em redes, interfaces homem-máquina e inteligência artificial [40].

Entender o contexto e atuar conforme, requer aprendizado. Muitas técnicas de *machine learning* foram desenvolvidas e adaptadas para tarefa de compreensão do contexto. Com o crescente número de sensores e dispositivos, com uma taxa cada vez maior, a geração e transmissão de dados na IoT se torna um ponto crítico de atenção. Coletar, gerenciar, processar e analizar estes dados requer novos métodos e técnicas para tratar o volume, variedade e veracidade de forma distinta as tecnologias tradicionais [40].

2.3 SEGURANÇA NA IOT

O ponto inicial da segurança na IoT se faz nos objetos. Estes geralmente estão, na maior parte do seu tempo de vida útil, desacompanhados - portanto suscetíveis a acessos físicos não-autorizados. Conforme seção 2.1, o ponto seguinte é a comunicação, que consiste majoritariamente sem-fio, facilitando escutas. As redes IoT, em sua maioria, são redes de larga escala, de cobertura ampla e totalmente descentralizadas, tornando-as predispostas a sofrer ataques em múltiplos pontos: indisponibilização da rede em si, transmissão de dados maliciosos/alterados na rede, avariação física dos objetos, acesso de dados pessoais/sensíveis (através das escutas), captura ou controle dos objetos, são alguns exemplos [14, 6, 10, 22].

Sem garantias de uma cobertura total com confidencialidade, autenticidade e privacidade, será improvável que os principais envolvidos (ou *stakeholders*) irão adotar a IoT plenamente. Portanto, utiliza-se a criptografia com uma das técnicas para garantir a confidencialidade, e em conjunto com outros mecanismos, a integridade e autenticidade dos dados, para tornar a comunicação resiliente contra ataques externos. Entretanto, a IoT não somente tem os mesmos desafios a serem superados das redes de sensores, dispositivos móveis e Internet, como tem

[3] *Ambient Intelligence*

requisitos distintos em privacidade, autenticação, controle de acesso, gerenciamento, entre outros. Geralmente os dispositivos são de 8-*bit* ou 16-*bits*, gerando uma necessidade de protocolos e mecanismos de criptografia melhor adaptados a esta realidade. Por exemplo, um problema consiste no caso de RFID o tamanho das senhas dos dispositivos são, em sua maioria, pequenas. E além, há o desafio de gerenciar senhas/dispositivos de várias entidades (possivelmente até na ordem de bilhões), tornando-se uma tarefa complexa e custosa [22, 10, 9, 11].

Devido à limitação, os dispositivos necessitam de cooperação para resiliência dos serviços ofertados. Para tanto, três pontos chaves são necessários nos objetos da IoT: serem seguros por padrão, aptos a entender a rede e seus serviços e resilientes a falhas na rede e ataques. Mecanismos de detecção (IDS), prevenção de intrusos (IPS) e serviços de recuperação, auxiliam na resiliência geral, prevenindo os ataques ou até sacrificando a performance dos serviços, sem indisponibilizá-los. Essa infraestrutura segura e auto-gerenciável é outro requisito fundamental da IoT, já que a grande maioria dos usuários não detém conhecimento técnico suficiente para tomar decisões sobre a segurança da mesma [22, 9, 39].

Vários conceitos são alicerces para prover segurança na IoT, geralmente combinam-se pra conseguir garantir: confidencialidade, privacidade, confiança, identificação e governança (sendo a última uma aglutinação de: políticas, auditoria, conformidade e rastreabilidade) [14, 22].

2.3.1 Privacidade

O registro dos eventos no contexto da IoT levanta muitas questões de privacidade, pois os dados coletados podem ser usados de forma imprevista (vide caso *Strava*, já apresentado) com efeitos no usuário final. As pessoas tem preocupações justificadas sobre a privacidade, pois, uma vez gerada a informação, existe uma grande probabilidade da mesma ser retida indefinidamente, negando o direito ao "esquecimento"- tema já regulado pelo parlamento europeu, por exemplo. Existem algumas iniciativas que propõem formas de realizar a tarefa de controle dos dados gerados, um exemplo é o P3P da W3C [10, 6, 41, 14, 39, 42].

Outro caso são de áreas em que há sensores monitorando, e há um desafio no conflito de direitos, já que indivíduos podem adentrar nestas áreas e não saberem quais informações estão sendo coletadas sobre si. Caso haja alguma proteção prevista, pode-se haver conflito entre a coleta e a privacidade dos envolvidos. Um domínio extremamente sensível à privacidade é a área da saúde, pois dados vazados tem impacto diretamente na vida. A disponibilidade destes dados criou um efeito *Big-Brother* que, caso não haja uma regulamentação, pode tornar-se uma distopia. Portanto, a privacidade deve estar desde o projeto, além de haver transparência e gerenciamento dos dados sensíveis. Assim, para efetivar a privacidade na IoT, deve ser possível aos indivíduos determinar quando outrem pode coletar e usar informações pessoais. Para um consentimento bem informado e livre, deve-se: entender o que está sendo feito, entender o contexto e uma ação afirmativa [14, 6, 43].

Algumas técnicas para cifragem dos dados e focadas em IoT já foram propostas: *Attribute-Based Encryption* (ABE), *Key-Policy Attribute-Based Encryption* (KP-ABE) e *Cipher-text-Policy Attribute-Based Encryption* (CP-ABE). Outra proposta, que agrega assinatura, é o esquema de *Attribute-Based Signature* (ABS). Entretanto, algumas tecnologias já utilizadas, podem ser usadas para melhorar a privacidade na IoT, tais como: *Virtual Private Networks* (VPN), *Transport Layer Security* (TLS), *DNS Security Extensions* (DNSSEC), *Onion Routing* (TOR, por exemplo) e *Private Information Retrieval* (PIR). Entretanto, cada opção agrega uma sobrecarga (pode ser na CPU, memória, ou até na latência de rede) que, em determinados contextos, pode tornar o uso impraticável. Todas estas técnicas e mecanismos dependem dos sistemas de autorização para avaliar acessos. Sistemas de autorização checam quando um usuário autenticado tem as permissões necessárias para o seu pedido em um recurso [8, 39, 43].

2.3.2 Confiança

A confiança é um conceito complexo o qual não há consenso em sua definição, porém é de extrema importância para definição de ajustes e comportamento dos objetos e do sistema no geral. Na segurança da IoT, qualquer interação entre objetos, recai na confiança direta ou indireta. Cada objeto deve mensurar a confiança em outros objetos baseando-se em credenciais, propriedades e outras informações para avaliar o compartilhamento de dados. Assim, a confiança tem uma relação direta com o gerenciamento de identidade e com o controle de acesso. Para tal, o esquema de gerenciamento de confiança deve ser distribuído e dinâmico, além de garantir um uso em que nenhuma confiança é definida *a priori*. Nota-se, inclusive, que os objetos não existem de "geração espontânea", geralmente estes fazem parte de grupos, tem localização em um contexto de aplicação, e pertencem a alguma entidade, auxiliando no gerenciamento da confiança [9, 22, 39].

Estas informações ao serem usadas adicionam um componente "social" aos objetos. As interações "sociais" que os objetos têm entre si, consideradas são: amizade (entre os donos dos objetos), propriedade (qual objeto pertence a quem) e pertencimento (qual objeto pertence a qual comunidade). Com isso mapeia-se alguns parâmetros para avaliar a confiança entre estes (conforme as pessoas já fazem), através de: honestidade, cooperação e interesse mútuo da comunidade. Paralelamente, há os ataques (igual as pessoas), como: difamação (um objeto ataca a rede dizendo que outro objeto é um atacante ou malicioso), auto-promoção (um objeto diz que é um roteador ou ataques *wormhole*) e bajulação (objetos maliciosos promovem-se para ganhar boa reputação e conseguir "dominar" a confiança da rede). Nota-se outro modelo de confiança, *Fuzzy Trusted Based Access Control* (FTBAC). Este modelo consiste em três camadas: dispositivos (a comunicação entre estes), pedidos (basicamente um banco de dados de SIGINT) e controle de acesso (o qual realiza o processo de decisão gerando um cálculo de confiança baseado no menor privilégio). A literatura apresenta outras propostas (como P2P, *Verifiable Caching Interaction Digest* - VCID, etc..), porém uma solução que suporte o volume de dados, interoperabilidade e dinamismo necessários, ainda está em aberto. A confiança é outro componente que permeia sistemas de autorização. Acessos e permissões partem de uma relação de confiança [39, 22].

2.3.3 Confidencialidade

A confidencialidade é um ativo importante no ecossistema da IoT, pois os dados e metadados são o valor central para adoção desta, e, devido ao impacto no negócio, estes dados podem representar ativos a serem resguardados dos competidores ou com valor na informação em si. Para garantir a confidencialidade, há dois impeditivos que se conflitam: escalabilidade e flexibilidade em tempo real. A escalabilidade é um desafio já apresentado, e a flexibilidade em tempo real, se faz, principalmente, em aplicações de políticas de acesso e autenticação durante o uso dos dispositivos, que inclusive podem mudar durante o uso. A relação ocorre entre duas entidades: detentores e coletores de dados, sendo os detentores pessoas e coisas, gerando dados e metadados. Os coletores são as entidades que vão agregar estes dados, precisando identificar e autenticar os detentores, para legitimação dos dados gerados [9, 39].

Para atingir o objetivo da confidencialidade, se faz necessário o uso de algum protocolo para o gerenciamento de chaves (KMS). Estes são classificados em 4 categorias de *frameworks*: conjunto de chaves (*key pool*), matemático, negociável e chaves públicas. Algumas propostas promissoras foram avaliadas na literatura, porém, até o momento, não há um padrão *de facto*. Geralmente a aplicação de um sistema se faz com auxílio de algum controle. Várias propostas existentes auxiliam nesta tarefa, moldando-se para o uso na IoT, conforme observa-se na seção 2.3.5. Importante frisar que as questões de usabilidade de qualquer uma das propostas devem

ser entendidas por um público leigo - portanto suscetível a interpretações e decisões inseguras [9, 39].

2.3.4 Identificação

Na IoT, cada objeto deve ser identificável, e dependendo do contexto, pode ser necessário o que seja unicamente ou que pertença a uma classe específica. Essa identificação pode ser atingida via duas classes: (1) identificação física: através de um rótulo (por RFID, QR code, etc); e (2) identificação lógica: que o próprio objeto anuncie-se. Estas duas classes não são excludentes, podendo ser usadas em conjunto [9].

Um problema que ocorre na identificação de objetos em IoT consiste no ataque de *proxy* (também conhecido como *man-in-the-middle* - MITM). Assim um atacante pode realizar uma identificação somente repassando os dados de um nó a outro, fingindo ser uma das pontas da comunicação. Note que, independentemente se os dados estão criptografados ou não, este ataque pode ser realizado. Por isso é necessário um conceito de identidade bem definido, que será a base para as decisões e usos nas políticas e acessos. Um ponto inicial consiste na federação do sistema de identificação. Assim, dada uma circunstância, o objeto pode definir quais dados pretende compartilhar (baseado em qual ambiente federativo se encontra, ambiente físico, requisitante, etc..), combinando informações como: o que sou, o que sei e o que tenho. Outra possibilidade são as identidades "sombra" (ou *shadow identity*), que consiste na noção que os objetos do usuário agem em seu nome, porém sem armazenar a sua identidade - identificando implicitamente o requisitante [6, 14, 9].

2.3.5 Controle de Acesso

O aspecto chave de segurança é o controle de acesso - decidir se irá ou não honrar um pedido. Portanto o controle de acesso é o mecanismo que garante que os recursos de um sistema somente serão acessíveis a usuários autorizados e inacessíveis a usuários não autorizados. O processo pode ser desmembrado em quatro partes[44]:

1. **Identificação**: auditoria da responsabilidade das ações executadas;
2. **Autenticação**: meio usado para provar o direito de uso de uma identidade, papel ou prova de posse de atributo(s);
3. **Autorização**: meios de expressar a(s) permissão(ões);
4. **Acesso**: decisão de honrar ou não pedido(s).

É comum aglutinar duas ou mais destas em um modelo de controle de acesso. Formalmente o controle de acesso consiste nas limitações das interações entre objetos e sujeitos em um sistema de informação. Os principais modelos de autorização, abaixo listados [45, 46, 47, 44, 48, 49]:

- ***Mandatory Access Control* (MAC)**: Um dos primeiros modelos de controle de acesso especificado foi o controle de acesso compulsório. Este é um modelo que restringe os acessos aos objetos baseado na confidencialidade da informação contida nos objetos e autorização formalizada de sujeitos para acessar a referida informação. Este mecanismo é projetado em níveis, com camadas de acesso vinculadas aos rótulos aplicados. Sujeitos com maior acesso podem alterar os rótulos dos objetos até a amplitude do nível que se encontra. *SELinux* e *AppArmor* são exemplos de implementações MAC.

- ***Discretionary Access Control* (DAC)**: Apresentado em conjunto com o MAC, o controle de acesso discricionário tem a característica que restringe o acesso a objetos conforme a identidade dos objetos ou grupos os quais pertencem. Uma forma de realizar o DAC consiste no uso de lista de controle de acesso (*Access Control Lists* - ACLs). Estas consistem no mecanismo de controle em que cada objeto tem uma lista de permissões de acesso para cada entidade. As permissões podem ser explícitas ou implícitas para os distintos modos de acesso (constantes em uma matriz, por exemplo). Sistemas de arquivos POSIX, roteadores e até alguns bancos de dados implementam ACLs, por exemplo. Um esquemático do DAC é apresentado na Figura 2.5(a).

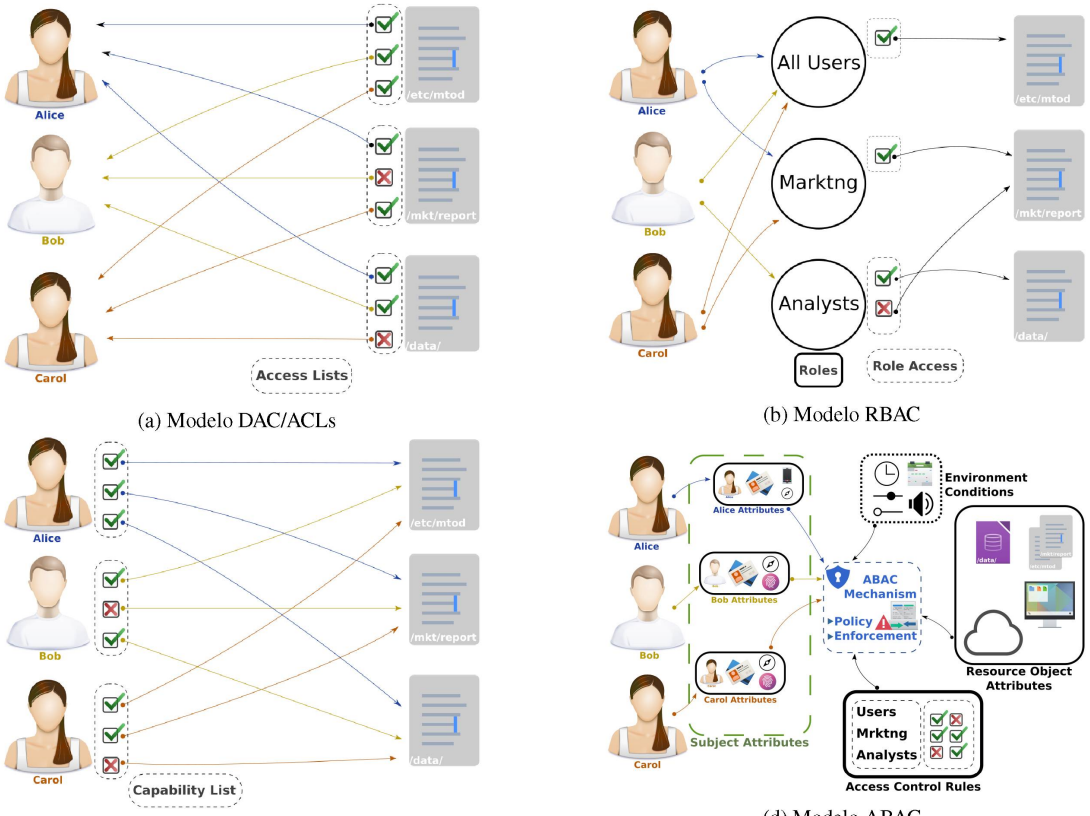
- ***Role Base Access Control* (RBAC)**: O controle de acesso baseado em papéis define um conjunto de usuários com respectivos papéis associados. São os papéis que definem o conjunto de acessos a objetos ou informações. Portanto assinala-se um papel para o(s) usuário(s) herdar(em) os privilégios autorizados constantes no primeiro. A principal diferença entre RBAC e DAC é o caso em que usuários não podem realizar mudanças nos objetos por conta própria - diferentemente do DAC/ACLs. O próprio RBAC é uma forma de MAC, mas não é baseado em requisitos de segurança multinível. O esquemático do RBAC é apresentado na Figura 2.5(b).

- ***Capability Based Access Control* (CapBAC)**: Uma habilidade (alguns sistemas classificam como chave ou *token*) é um bilhete comunicável e imutável de autoridade. Este consiste em um valor que referencia um objeto agregado ao conjunto de seus acessos associados. Assim, a habilidade é definida como um objeto protegido, o qual, estando em posse por um sujeito, autoriza este sistema a habilidade de interagir com um objeto de certas formas. Estas interações podem ser: leitura dos dados associados com o objeto, modificação do objeto, execução dos dados do objeto como processo, entre outros acessos possíveis. Logicamente, a habilidade consiste na referência que identifica unicamente um objeto particular e seu(s) respectivo(s) conjunto(s) de acessos. O CapBAC é visto na Figura 2.5(c).

- ***Attribute Based Access Control* (ABAC)**: Neste modelo o sujeito e o objeto se identificam através de atributos associados a características dos mesmos. O sujeito recebe as permissões apropriadas de acesso ao objeto de acordo com os atributos no momento do pedido. Portanto, descreve-se o emissor e o pedido de acesso através de atributos, incluindo algumas restrições que inerentes via atributos de ambiente (por exemplo). Com ABAC habilita-se para todas as entidades envolvidas definirem permissões de acessos baseadas em qualquer característica de segurança relevante (os atributos). Vê-se a diferença do ABAC com os outros modelos no esquemático apresentado na Figura 2.5(d).

A autorização gera um desafio de alcançar uma segurança horizontal quando aplicada entre domínios. No caso específico de ABAC, este resolve o problema da grande quantidade de regras através da possibilidade de usar combinação de vários atributos envolvendo um pedido específico, gerando uma política dinâmica (e potencialmente reduzindo o número de regras estáticas associadas com cada tipo de dispositivo/recurso). Um número potencialmente alto de atributos que são necessários para entender e gerenciar uma política dinâmica ainda é um desafio do mesmo. Uma dificuldade persiste: princípio do mínimo privilégio [51, 48].

Nos modelos ACL, RBAC e ABAC têm-se uma entidade central que gerencia a política, porém esta consiste num ponto único de falha (SPoF). O modelo CapBAC mitiga esse SPoF com uma abordagem totalmente distribuída, mas sem levar em consideração a limitação de



(a) Modelo DAC/ACLs

(b) Modelo RBAC

(c) Modelo CapBAC

(d) Modelo ABAC

Figura 2.5: Principais modelos de autorização [48, 46, 50].

recursos dos dispositivos de IoT. A geração de *tokens* realizada neste modelo facilita a delegação, revogação de acesso e granularidade. Entretanto, isso traz um ponto de atenção no CapBAC: os usuários necessitam de conhecimentos de segurança, pois estes são responsáveis pela geração de *tokens* para outros usuários/dispositivos - autorizando ou revogando acessos. Conforme visto na seção 2.3.3, esta tomada de decisão dos usuários pode ser insegura. Por último, modelo CapBAC não leva em consideração o uso do contexto [51, 48, 52].

A família de modelos de controle de uso conhecida como UCON$_{ABC}$ destoa dos modelos apresentados anteriormente. Este modelo de controle de uso integra **A**utorizações, o**B**rigações e **C**ondições. Os elementos $_{ABC}$ são combinados para realizar os privilégios de sujeitos e objetos. Além disso, UCON introduz duas novas funcionalidades: continuidade e mutabilidade. UCON possibilita que os valores de atributos dos sujeitos podem ser alterados não somente após o controle de acesso, mas durante. UCON também oferece características de alto dinamismo, flexibilidade, granularidade e escalabilidade, porém com desvantagens de não ter funcionalidade para administração, sem delegação, alta complexidade e baixa usabilidade. Ainda faltam trabalhos que implementam o uso de UCON para dispositivos com capacidade reduzida [53, 54, 52].

O desafio é executar a autorização em dispositivos limitados. Tornando o problema de manter políticas distribuídas e atualizadas - além da necessidade das revogações, quando necessárias. Porém há falta de propostas que seja amigável ao usuário final para configuração de políticas [43].

2.4 TRABALHOS CORRELATOS

Conforme apresentado na seção 2.3.5, existem várias propostas de arquiteturas e modelos de controle de acesso. Cada qual com suas características que se encaixam para cada tipo de uso. Já visto na seção 2.3.1, mostrou-se a importância da privacidade. Estes dois conceitos devem caminhar juntos para um controle de acesso fácil e confiável por parte dos usuários. A seguir alguns trabalhos correlatos e suas principais características.

2.4.1 *Software Defined IoT Security Framework*

Este trabalho propõe um *framework* seguro definido por *software* (SD) para IoT. Este modelo baseado em *Software Defined Network* (SDN) tem a característica de configurar políticas e regras por toda a rede de forma consistente. Em SDN, a separação dos planos de controle e dados é vital para habilitar políticas de acesso dinâmicas, flexíveis e configuráveis. Este *framework* engloba tanto autenticação quanto autorização [55].

A hierarquia proposta neste modelo é em camadas, com as seguintes características:

1. *Device Layer*: A camada de dispositivos consiste em um conjunto heterogêneo de dispositivos e tecnologias com diferentes capacidades. Nesta camada, cada conjunto de dispositivos têm um certo nível de dependência nos esquemas de autenticação suportados e a aplicação/serviço;

2. *Access Network Layer*: A camada de acesso à rede consiste em um conjunto de pontos de acesso (*relays*), os quais estão equipados com SDN. Estes dispositivos de acesso são gerenciados pela camada de controle de acesso;

3. *Access Control Layer*: A camada de controle de acesso consiste nos pontos de controle. Estes pontos inteligentes incluem controladoras que gerenciam as camadas heterogêneas de acesso e dispositivos;

4. *Core Network Layer*: A camada central da rede consiste em um conjunto de elementos híbridos que conectam os conjuntos de redes de acesso. Essa camada é controlada pelo gerenciador da SDN;

5. *Core Control Layer*: A camada central de controle é responsável por gerenciar a camada central de rede. Possui o nível mais seguro e tem o papel central no controle de acesso. Tem a visão de todas configurações, autenticações, regras de acesso, etc.. da rede. Pode ter comportamento reativo para mudanças dinâmicas na política;

6. *Application Layer*: A camada de aplicação é o conjunto de aplicações que rodam sobre a SDN. Devido à virtualização da rede e uma miríade de aplicações, se faz necessário um controle de acesso rígido.

Conforme a figura 2.6, este *framework* apresenta um modelo de controle de acesso MAC/DAC pela visão do controlador central (item 5) e um controle de acesso RBAC/CapBAC combinado com ABAC para a camada de acesso ao meio (item 2). Todo pedido de acesso realizado por dispositivos é enviado para o ponto de acesso mais próximo, o qual busca pela respectiva regra associada. Neste modelo é considerado que o pedido é processado por aplicação e por sessão [55].

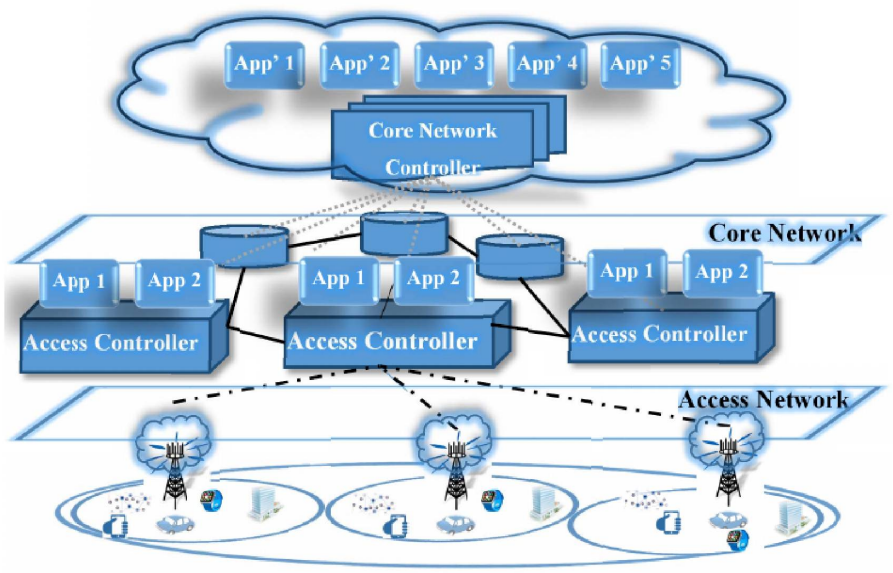


Figura 2.6: Arquitetura do *framework* SDN proposta [55].

2.4.2 *IoT@Work*

O mecanismo apresentado no *Iot@Work* consiste num modelo CapBAC [48]. Conforme visto na seção 2.3.5, este modelo é baseado em habilidades, que aqui se denominam *tokens* (conforme visto na seção 2.3.5). Estes *tokens* demonstram a posse de uma autorização específica.

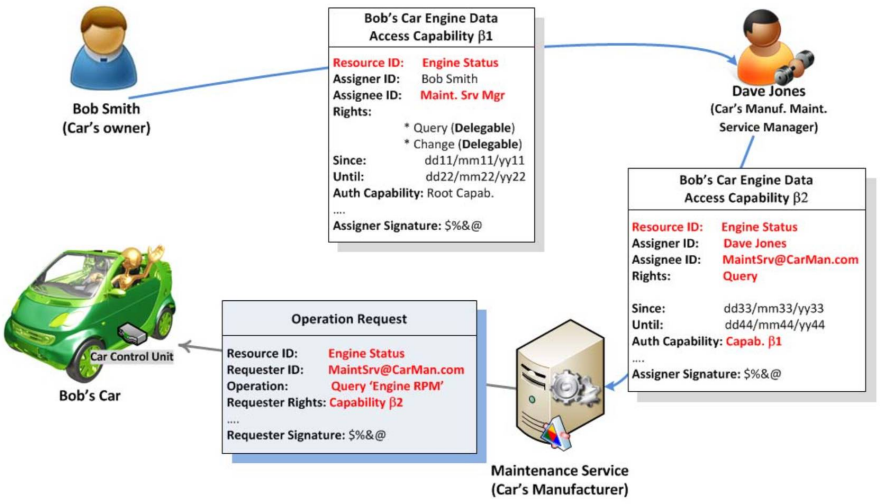


Figura 2.7: Cenário de uso do CapBAC.

Um exemplo apresentado na figura 2.7 consiste no uso do modelo para o controle de acesso das informações e serviços do carro de *Bob*. Neste, *Bob* autoriza o mecânico, via *token*, o acesso à informações do motor. Por sua vez o mecânico envia o pedido de acesso, incluindo a autorização do *Bob*, para a fabricante, que encaminha o pedido de operação para o veículo.

Na figura 2.8 é apresentado outro exemplo: interação entre duas pessoas e duas empresas. *Alice* autoriza, via o envio de um *token* específico, *Cartoonia* (empresa de *Bob*) a realizar o serviço "A". *Bob* realiza a validação cruzada com o servidor da empresa *Acme* (para garantir que *Alice* pode realizar pedidos em nome da empresa) e pede para sua empresa (*Cartoonia*)

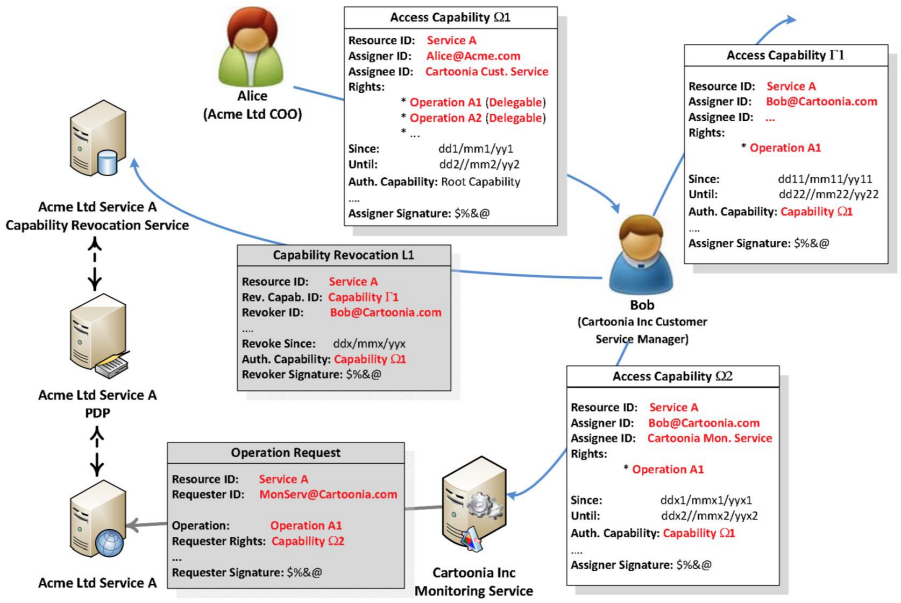


Figura 2.8: Elementos do modelo de autorização CapBAC [48].

referendar o seu pedido. Este exemplo mais completo envolve os elementos do modelo CapBAC, listados a seguir:

- *Recurso*: Objeto do *token*. Contém a única restrição de que este recurso seja identificável de forma inequívoca e realize uma ação num objeto;
- *Authorization Capability* (AC): Contém os detalhes dos privilégios, é o recurso que garante os direitos, identidade responsável pela autorização, e todas informações necessárias;
- *Capability Revocation* (CR): Contém a revogação de privilégios;
- *Service/Operation Request* (SR): Pedido de serviço/operação conforme especificado pelo provedor de serviço, de maneira que referencia ou identifica, de forma única, uma habilidade;
- *Resource Policy Decision Point* (PDP): Serviço encarregado de gerenciar, validar e decidir sobre os pedidos de acessos a recursos;
- *Resource Manager* (RM): Serviço que gerencia os pedidos a um recurso identificado;
- *Revocation Service* (RS): Serviço responsável por gerenciar as revogações de acesso;

Observa-se neste modelo que a principal desvantagem consiste na capacidade de emitir *tokens* de acessos para todos os objetos. Ainda se faz necessária a padronização da estrutura de dados contida nos *tokens*. Um ponto de atenção neste modelo consiste em não promover a privacidade como requisito, ponto em aberto para complementação/futuras extensões [48].

2.4.3 *Community Capability-Based Access Control*

No trabalho [51], é apresentado uma estrutura de controle de acesso focada em comunidade. Neste a comunidade é definida como um agrupamento de dispositivos com as seguintes características: compartilhamento de objetivos em comum, um conjunto de sistemas tendo uma hierarquia conforme sua capacidade e uma política comum. Esta estrutura é denominada COCapBAC - *Community Capability-Based Access Control*. Os componentes deste modelo estão apresentados na figura 2.9:

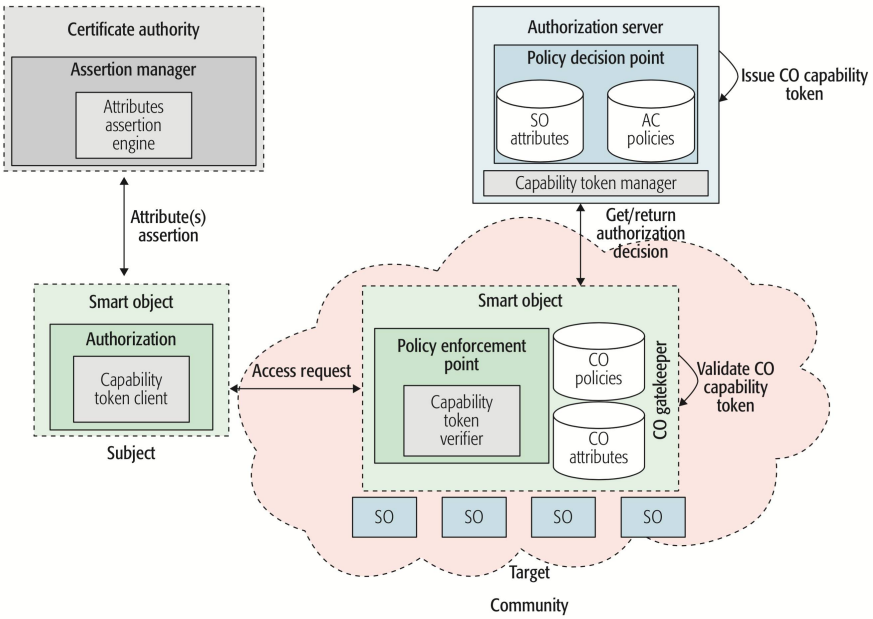


Figura 2.9: Modelo de autorização COCapBAC[51].

Conforme a figura 2.9, os componentes têm as seguintes características:

- *Smart Object* (SO): O objeto inteligente consiste no dispositivos o qual faz parte da comunidade da qual compartilha um objetivo em comum;
- *Authorization Server* (AS): O servidor de autorização é responsável por conceder a autorização de acesso à comunidade. Para isso registra os atributos dos dispositivos e as políticas da CA. O AS geralmente é o PDP;
- *Certification Authority* (CA): Autoridade certificadora responsável pela emissão de *tokens* de autorização;
- *Policy Decision Point* (PDP): Atua como um ponto de decisão de política, que analisa e toma as decisões de autorização;
- *Policy Enforcement Point* (PEP): Responsável pelo cumprimento das decisões da comunidade enviadas pelo PDP. Também tem o papel de validar se um *token* foi alterado por uma terceira parte. Este faz parte do CG;
- *Comunity Gatekeeper* (CG): Responsável pelo cumprimento de uma decisão de autorização feita pelo AS dentro da comunidade. Para tal, registra os atributos dos dispositivos que fazem parte da comunidade bem como a política desta comunidade, dados providos pelo AS;

- Comunidade (CO): Consiste no grupo de dispositivos e serviços que compartilham objetivos em comum e são gerenciados pelo mesmo AS;
- Capacidade (*token*): Consiste em uma estrutura de dados que contém um conjunto de privilégios, os quais são emitidos e assinados pelo AS e validados por um CG.

2.5 CONSIDERAÇÕES

Observa-se que a explosão de negócios e serviços com IoT cresceu e irá crescer em ordens de grandeza nos próximos anos. Outro ponto a se notar é a elasticidade inerente da IoT, pois os números de dispositivos geralmente ficam nas grandezas de milhares, até milhões, de unidades. Soma-se também, a limitação de computação destes dispositivos, tornando o uso de *cloud* e *fog computing* um complemento.

Ademais, a interoperabilidade traz a necessidade de um componente crucial: segurança. Todos estes desafios devem partir da premissa de comunicação segura. Observou-se que a pesquisa de segurança no contexto de IoT é ampla e infindável. O recorte apresentado aqui, se faz para um foco maior na área da proposta, porém não cobre outra infinitude de temas relevantes para o todo.

Este trabalho foca no aspecto de controle de acesso e uso do contexto como suporte a decisões de segurança, tema das seções 2.3.5 e 2.2. A seguir, descrever-se-á o protocolo, seu vocabulário e funcionamento.

3 PROPOSTA

Este trabalho propõe um protocolo de autorização para IoT baseado em atributos e informações de contexto, baseado no modelo ABAC. Conforme visto na seção 2.3.5, o modelo ABAC detém algumas características importantes para IoT: granularidade, escalabilidade, flexibilidade, interoperabilidade e suporte a delegação. O ABAC é composto de dois aspectos: o modelo da política e o modelo da arquitetura - o qual aplica a política. Desta forma, o modelo define quais acessos podem ser realizados via um determinado conjunto de atributos apresentados por um sujeito. Já as regras (política) especificam as condições as quais os acessos serão permitidos ou negados [52, 56].

Entretanto, o modelo ABAC apresenta algumas desvantagens de complexidade e falta de usabilidade. A forma proposta de tentar mitigar estas desvantagens se faz com um domínio e a figura de um gerenciador (que concentraria o *PDP/"PEP-Master"* do primeiro). Além deste, combina-se características similares à ReBAC em um componente social das interações entre os indivíduos, tornando-se um 4^o fator de segurança [52, 56, 57].

Esta estratégia consiste em utilizar as interações sociais para aumentar a usabilidade e computar riscos. Com isso, o gerenciador quantificará a confiança dos dispositivos participantes, analisando comportamentos e expectativas para saber se estão em conformidade. No que tange o risco, não se caracteriza um RAdAC pois este risco não se aplica granularmente em cada pedido de acesso e sim na combinação de dispositivos + informação acessada. Esta mudança tem o objetivo de minimizar a sobrecarga de processamento disponível em dispositivos limitados. Assim realiza-se uma análise de risco, classificando os dispositivos em graus de comprometimento dos mesmos, gerando uma confiança entre os membros do domínio. O principal objetivo do protocolo é aplicar um modelo de autorização que aproveite informações do ambiente e a interação entre dispositivos de forma a facilitar o uso destes - sem sacrificar a segurança. Esta facilidade se propõe com a capacidade de autoconfiguração segura e focada na privacidade durante o uso.

3.1 ARQUITETURA

Os principais componentes e termos do protocolo estão listados a seguir:

Domínio (*Realm*): é o conjunto lógico do qual fazem parte um ou mais dispositivos. Neste conjunto denota-se uma confiança e um objetivo comum entre os membros. Há o suporte da federalização de domínios, assim cada dispositivo pode participar de um ou mais domínios. Há um domínio "base", que serve para descoberta dos domínios existentes na rede local;

Gerenciador (*Manager*): é o dispositivo responsável pelo gerenciamento do domínio. Durante sua eleição, sempre será escolhido aquele dispositivo que tiver a melhor disposição de carga de trabalho. Devido a isso, tende a ser o dispositivo de maior poder computacional. O gerenciador também tem o papel de entidade certificadora-raiz do domínio (podendo delegar sub-certificadoras para escalabilidade). Será responsável por realizar várias tarefas, como por exemplo:

1. Aceitar ou não novos dispositivos no domínio;
2. Definir quais desafios cada dispositivo deverá responder;
3. Definir quais dispositivos irão participar de um desafio;

4. Definir a cadeia de comando e sucessão;
5. Atualizar a política e os pontos de assertividade (PDP e PEP);
6. Atualizar e distribuir os coeficientes sociais e de risco;
7. Definir quantas identidades cada dispositivo terá;
8. Autoridade Certificadora (CA) do domínio - com subCAs;
9. Gerenciar a relação de confiança com outros domínios;
10. Prover serviços de *fog/cloud*, conforme implementação.

Dispositivo (*Thing*): todos os outros dispositivos que fazem parte do domínio. O gerenciador pode elencar "tarefas", com o intuito de diminuir sua carga computacional, para os membros do domínio. Todos os dispositivos do domínio devem auxiliar na resposta de desafios. Todos criam uma base "social", com observações da última vez que viu outros dispositivos. Além disto, essa base também serve para cada dispositivo acompanhar o "comportamento" dos outros, de forma a criar um "tecido social" e gerar um coeficiente de confiança;

Confiança (*Trust*): coeficiente com o qual cada dispositivo classifica os outros membros do domínio. Composto da taxa de desafios, comportamento (caso o dispositivo mantenha operações conforme previsto), risco da informação vazada, entre outros. A confiança é mapeada em uma matriz com os coeficientes de todos os membros do domínio. O coeficiente é configurável via política e conforme aplicação;

Membros (*Members*): são os dispositivos do domínio. Há duas classes: total e parcial. Os membros parciais são todos os dispositivos que estão no processo de entrada no domínio. Neste processo, um número limitado de tarefas e interações são permitidos aos membros classificados como limitados. No outro caso, membros totais consistem nos dispositivos que realizaram o processo de entrada no domínio e já atingiram o coeficiente mínimo de confiança;

Reino (*Kingdom*): é meta-domínio. É a federalização dos domínios. Para formar um reino, os gerenciadores devem realizar um processo de confiança mútua.

Cabe ressaltar que neste trabalho os termos **Domínio** e *Realm* se referem ao mesmo conceito. O Contexto refere-se à representação lógica do ambiente observado. Na Seção 3.2 é apresentado o escopo deste trabalho, com suas premissas e mais definições pertinentes.

Espera-se que o gerenciador tenha uma capacidade de processamento adequada conforme suas tarefas, sendo como exemplo de capacidade: *smartphone*, computadores portáteis e de mesa e alguns dispositivos com recursos compatíveis com um *Raspberry Pi* (e dipositivos com capacidades de computação nesta categoria, tais como: *Orange Pi, Parallela, Banana Pi,* entre outros). Já a capacidade esperada de um dispositivo do domínio se inicia ao nível de um *Arduino* (e correlatos: *vCore e microduino*), limitado e com possibilidade restrita tanto em memória, quanto em processamento.

Na Figura 3.1 observa-se um exemplo de domínio único com os elementos da arquitetura: 1 gerenciador e 5 dispositivos. Também é possível visualizar que cada dispositivo pode criar a quantidade de identidades que o gerenciador indicar. Demonstra também que cada dispositivo contém uma certa capacidade de processamento, sensores e atuadores.

Na Figura 3.2 estão apresentados 4 exemplos de uma progressão da população de dispositivos de um domínio. Conforme a figura 3.2(a), no início um dispositivo (no caso, uma

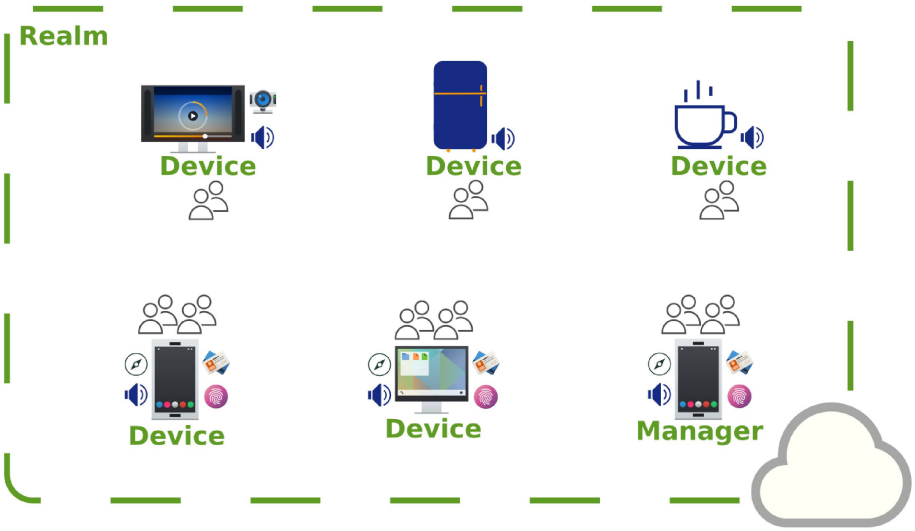
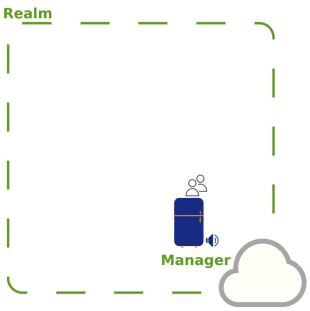
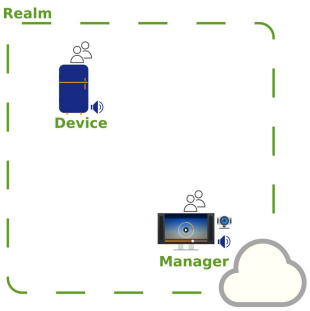


Figura 3.1: Exemplo da arquitetura com os elementos citados.

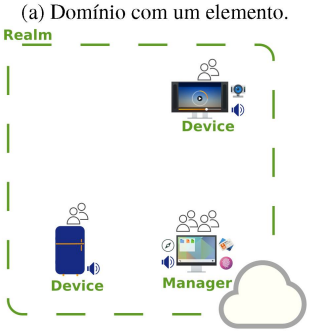
geladeira) entra no domínio e se torna o gerenciador do mesmo. Mesmo com capacidade limitada, já que esta é única neste domínio, neste momento. No momento seguinte (figura 3.2(b)), uma televisão entra no domínio. Após a eleição do novo gerenciador, a televisão (com maior capacidade) se torna o novo gerenciador. Na sequência (figura 3.2(c)) o processo de eleição ocorre novamente e o computador é eleito o novo gerenciador. Por último (figura 3.2(d)), o *smartphone* entra no domínio, ocorre a nova eleição, fazendo novamente o computador como gerenciador.



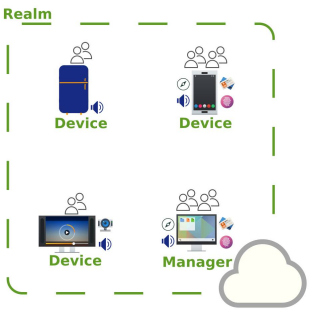
(a) Domínio com um elemento.



(b) Domínio após a entrada de um elemento com maior capacidade.



(c) Domínio após a entrada de outro elemento com maior capacidade.



(d) Domínio após a entrada do último elemento.

Figura 3.2: Exemplo do populamento de dispositivos em um domínio.

Cabe frisar que sempre ocorrerá uma eleição por dois motivos: novo dispositivo entrou no domínio ou o gerenciador atual saiu do domínio (voluntariamente, via *timeout* ou por comprometimento). O processo de entrada no domínio contém subtarefas condicionantes de entrada. Também é definido toda a cadeia de comando, assim pode-se ter a flexibilidade de ter

timeouts distintos para indisponibilidade do gerenciador. Um destes (chamado de *blink timeout* - BT) consiste em uma intermitência ou um período curto de indisponibilidade do gerenciador, fazendo com que a linha de sucessão distribua e execute as tarefas dele. Já o *timeout* seguinte (chamado de *long timeout*- LT) consiste em outro período o qual o gerenciador é dado como ausente do local, sem bateria ou energia; e sem previsão de retorno, necessitando realizar a nova eleição.

3.2 ESCOPO

Algumas premissas e decisões influenciam nos objetivos e comportamento esperado do protocolo. Abaixo estão listadas as premissas deste trabalho:

Modelo do sistema: Assume-se que o usuário tenha um *smartphone* e outro dispositivo qualquer, cada um com, no mínimo, sensores complementares. Essa característica de "complementação" entre sensores de dispositivos distintos será explicada de maneira detalhada na seção 4.1. Presume-se dispositivos comuns com acesso wifi. Os casos de uso previstos neste momento (mas não limitados) são de uma casa com variedade de dispositivos capazes;

Usabilidade: O objetivo do protocolo é facilitar a configuração dos dispositivos, sem perder a configuração segura "de fábrica". Usuário deverá necessitar de poucas interações (que acarretem em decisões) com o grupo de dispositivos;

Flexibilidade: as decisões de segurança podem ser alteradas conforme perfil de uso. Por exemplo o *timeout* de resposta de um dispositivo. Este dispositivo pode ter saído do ambiente, pode ter acabado sua bateria ou até mesmo desligado voluntariamente. Assim, dependendo da característica do ambiente, este intervalo de tempo deve ser ajustado para melhor performance e segurança;

Atributos: O protocolo se baseia em contexto e suporta a aquisição, representação, entrega e reação conforme as informações coletadas. Suporta o uso de ambos contextos: primário e secundário. São utilizados os seguintes atributos de contexto: luminância, câmera, LEDs, som e microfone, por exemplo. Há o componente "social", que é baseado na interatividade dos dispositivos presentes no ambiente. Além disso, suporta a configuração de atributos conforme a aplicação, já que há outras possibilidades de observação física: temperatura, pressão, giroscópio, força de sinal rede sem-fio e geolocalização, para citar alguns.

Cabe ressaltar que o comportamento dos dispositivos derivam das decisões políticas. Por exemplo o *timeout*: caso seja muito curto, oscilações de comunicação/sinal podem gerar falsos positivos; caso seja muito longo, os membros do domínio podem ter que esperar uma nova eleição para o retorno da normalidade deste domínio. Outro exemplo consiste na avaliação de riscos: caso haja uma maior "desconfiança", por padrão, os dispositivos que estejam sofrendo falhas em *hardware* e, consequentemente, faça leituras erradas no(s) sensor(es), podem ser classificados - erroneamente - como dispositivos comprometidos.

Conforme visto na seção 2.2, um modelo de autorização, ao se basear em contexto, deve suportar: aquisição, representação, entrega e reação, conforme as informações coletadas. Além disso, o contexto está fortemente acoplado na aplicação, mesmo com a possibilidade de contexto implícito em alguns modelos. Nesta proposta terá o uso de ambos contextos: primários e secundários. Contexto primário consiste na informação coletada sem utilizar contexto ou

qualquer dado agregado e/ou inferido. O contexto secundário consiste em qualquer informação que pode ser computada ou inferida de dados de contexto primário. Por exemplo, um serviço que faça a tradução dos pontos de latitude e longitude (localização) em informações de endereço, cidade, estado. Com isso infere-se qual ambiente o dispositivo atualmente se encontra [5].

3.3 COMPARATIVO

Em relação aos trabalhos correlatos apresentados na Seção 2.4 este trabalho apresenta uma mudança de abordagem, de centrada na identidade para uma abordagem centrada em atributos. Esta mudança de abordagem se dá, pois as identidades dos dispositivos agregam uma maior complexidade em grande escala, dificultando a manutenção. O gerenciamento de autorizações é semi-distribuído, ao contrário de uma visão otimista da literatura, que propõe um modelo totalmente distribuído, já que os dispositivos possuem capacidade limitada. Descarta-se uma abordagem centrada no usuário para uma abordagem auto-contida e automatizada. Por fim, os usuários definem as políticas de forma geral e, observando esta, os dispositivos formam as comunidades.

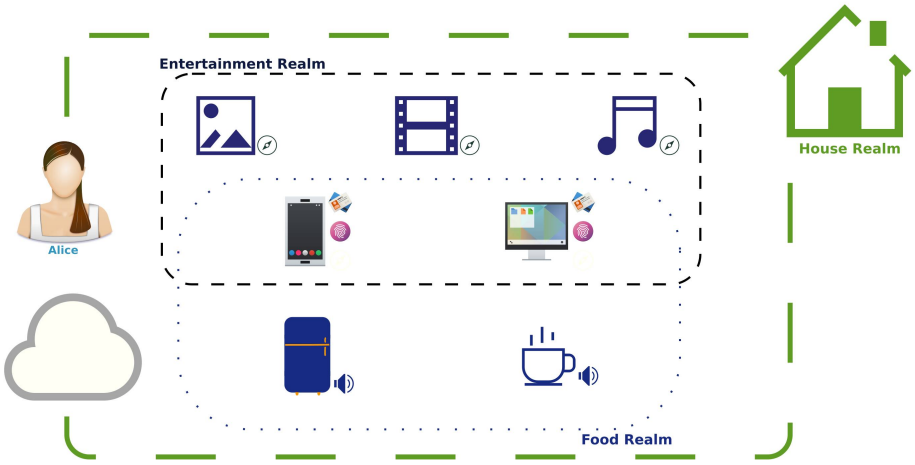


Figura 3.3: Esta proposta: exemplo durante uso com a distribuição dos dispositivos e 2 domínios.

Um comparativo das propostas apresentadas com este trabalho, está sumarizado na Tabela 3.1, a seguir:

Propostas	Controle	Agrupamento	Privacidade	Contexto
Salman[55]	RBAC+ABAC	Federado	○	○
Hussein[51]	COCapBAC	Comunidade	◑	○
Gusmeroli[48]	CapBAC	Federado	◑	○
Este Trabalho	ABAC+n-FA	Federado	●	●

Tabela 3.1: Comparativo de propostas de modelo de autorização.

Alguns trabalhos trazem a discussão do aumento da superfície de ataque em soluções de SDN. Devido ao rápido crescimento e uso, os gerenciadores das SDNs podem ser classificados como um Sistema Operacional em Rede (NOS). Caso estes gerenciadores forem comprometidos, podem causar problemas sérios, como *black hole routing* e geração de enlaces falsos [58].

3.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a proposta e respectivos componentes envolvidos. Apresentaram-se: Domínio, Gerenciador, Dispositivos, Confiança, Membros e Reino. Foram apresentados o que espera-se de características de *hardware* e *software* para os papéis no domínio.

Outro ponto abordado foi o escopo da proposta, com respectivos: modelo do sistema, usabilidade, flexibilidade e atributos. Por fim, foi apresentado um comparativo desta proposta com trabalhos correlatos.

No Capítulo 4, será apresentado o Modelo de confiança entre os dispositivos - descrevendo os desafios e seus usos - e, as fases do protocolo. No Capítulo 5, serão discutidos os aspectos de implementação, modelo de ameaças e respectivas mitigações, além do uso dos atributos e experimentos realizados.

4 MODELO DE CONFIANÇA ENTRE DISPOSITIVOS

Neste capítulo, apresenta-se o modelo de confiança do protocolo. Este modelo baseia-se no uso das informações de ambiente. Estas informações, observadas via sensores dos dispositivos, criam um contexto para os mesmos. Este contexto oferece a possibilidade de uso, garantindo que os dispositivos definam se estão ou não no mesmo ambiente físico. Assim geram-se perguntas sobre observações físicas que os dispositivos possam ler e verificar, chamadas no protocolo de desafios. A seguir são apresentados o conceito dos desafios e respectivos usos.

4.1 DESAFIOS DE CONFIANÇA

Os desafios são provas de que os dispositivos estão no mesmo ambiente (ou próximos fisicamente), mitigando a possibilidade de fraude. Os desafios devem ter dificuldade suficiente tal que dispositivos que não se encontram no mesmo ambiente não conseguirão responder os desafios corretamente ou em tempo hábil. Os desafios são baseados nos sensores que cada dispositivo possuir e podem ser combinados para geração de confiança. Exemplos das principais observações e respectivas características:

Luminância: consiste na quantidade de luz de um ambiente. Estes atributos têm um contraponto, pois tende a variar dentro de um mesmo cômodo/ambiente (por exemplo, o dispositivo está na sombra de outro ou em uma gaveta);

Câmera/LED: conjunto de uma câmera e LEDs controláveis pelo dispositivo. Para seu uso será necessário que exista um par de dispositivos distintos que os contenham. Cabe ressaltar que os LEDs devem estar no campo de visão da câmera;

Som: este consiste em um par também: *speaker* e microfone. Possui características similares ao item anterior - porém sem a limitação do "campo de visão";

Social: este é o atributo que envolve um coeficiente social baseado na relação entre os dispositivos durante o ciclo de vida no domínio.

A Tabela 4.1 apresenta alguns exemplos de atributos e respectivos usos nos desafios. Esta combinação de sensores traz maior segurança na relação dos dispositivos do ambiente, já que leituras complementares e votação de leituras diminuem a possibilidade de falsificação. Aumenta-se, com isto, o custo para falsificar o pertencimento ao ambiente. Quanto maior o número e diversidade de sensores no domínio, menor será a possibilidade de um dispositivo que não se encontra naquele ambiente garantir sua presença no domínio - aumentando a desconfiança dos outros membros.

No caso de falha de *hardware*, provocando a impossibilidade de responder adequadamente aos desafios, ocorrerá a penalização do dispositivo. À primeira vista, parece injusta esta tratativa, mas, se um dispositivo está falhando na leitura e interação com os outros membros do ambiente, é um indicativo de fim da vida útil do mesmo ou que o dispositivo não está no ambiente correspondente. Em qualquer dos casos, a desconfiança aumenta e, ao passar de um limiar, o dispositivo é removido do domínio por precaução.

O componente social, que é pervasivo ao domínio, consiste na matriz de riscos - repositório da confiança dinâmica. Ao registrar-se, cada dispositivo receberá os desafios do gerenciador conforme capacidades anunciadas do mesmo. Ao acertar os desafios, pontua-se

Tabela 4.1: Tabela de atributos e desafios

Atributo	Desafio Validação	Comentário
Temperatura	leitura/leitura	A validação de uma leitura de um membro do domínio se fará com a votação de todos os outros dispositivos;
Luminância	leitura leitura	A validação de uma leitura de um membro do domínio se fará com a votação de todos os outros dispositivos;
Som	emissão captura	Aqui começam os atributos que são complementares - microfone e um *speaker*/caixa de som, caso haja mais de um exemplar de microfone no domínio, há votação;
Câmera	LEDs captura	Neste caso, há a possibilidade de combinar dispositivos que tenham câmeras, com dispositivos que tenham possibilidade de emissão de códigos. Podem ser piscando LED's de uma sequencia específica, QRCodes, etc..
Social	observação votação	Cada membro contabiliza erros e acertos na comunicação e interatividade com outros membros do domínio. No momento em que há o *broadcast* dos coeficientes observados, cada dispositivo irá repassar os coeficientes que observou e receberá os coeficientes dos outros. Ao final o gerenciador classificará quais membros poderão sofrer limitações e/ou promoções;

positivamente e, o contrário, negativamente. Além disso, durante o registro, ao anexar os atributos do mesmo no pedido, o gerenciador os assinará para que a interação do dispositivo com outros membros do domínio ocorra. Por exemplo, ao receber de um novo membro um atributo assinado, de que este é uma câmera de segurança (CCTV), e este novo membro começa a realizar tentativas de logins, ou consultas sobre quais sons estão ocorrendo no ambiente, caracteriza um comportamento fora do previsto, portanto penalisado, já que pode indicar um comprometimento do dispositivo em si.

4.2 *SCORES* DE CONFIANÇA E RISCO

Confiança é um conceito de difícil definição, sendo influenciado por muitas propriedades mensuráveis e não-mensuráveis. O conceito de confiança cobre um escopo maior que segurança, tornando-a mais complicada e difícil de estabelecer, manter e garantir. Outro conceito relacionado à confiança é a privacidade, que é a habilidade de uma entidade determinar quem, quando e se uma informação própria será comunicada. Alguns parâmetros são usados para quantificar a confiança entre dispositivos: proximidade, comportamento, cumprimento, histórico, similaridade, interações, entre outros. Estes parâmetros auxiliam na confiança direta (comunicação entre dispositivos) ou indireta (dispositivos propagam sua confiança sobre outros) [59, 60].

O protocolo suporta o uso destes parâmetros através dos desafios e interações sociais entre os dispositivos para manter um coeficiente de confiança entre os mesmos. Este coeficiente é batizado de *Mafia Score* (MS). O MS é calculado levando em consideração os seguintes pontos: interatividade com outros dispositivos, comportamento, taxa de acerto dos desafios, histórico, terceiros e risco.

O risco pode ser definido como a possibilidade de perda ou prejuízo. Geralmente o risco é sobre algum evento que possa ocorrer no futuro e cause perdas. O risco é quantificável pela probabilidade de um incidente ocorrer e o impacto estimado do prejuízo deste incidente. Esta análise pode ser adaptável ou estática. A abordagem adaptável calcula dinamicamente durante o monitoramento do uso e ajusta novos valores de risco. A estática apenas calcula os riscos no início do uso - sem monitorar e ajustar conforme o ciclo de vida da informação [61, 62]. Ao participar de um domínio, o gerenciador envia para o dispositivo o coeficiente de prejuízo de ações *versus* informações. O gerenciador irá calcular conforme as características e política de privacidade configurada.

4.3 GESTÃO DE DOMÍNIOS DE CONFIANÇA

4.3.1 Ciclo de Vida do Dispositivo

Partindo do ciclo de vida de operação de um dispositivo, este observará o diagrama de estados da figura 4.1. Neste diagrama definem-se os seguintes estados de operação:

Externo (não autorizado): Estado inicial. O dispositivo faz um anúncio de busca de domínio e fica em espera até a resposta do(s) gerenciador(es) de domínio(s). O dispositivo já realizou a autenticação e está no ponto inicial de autorização. Todo dispositivo inicia as operações do ambiente neste estado;

Pendente: Este é o estado em que o dispositivo enviou o pedido de autorização ao gerenciador - anexado de seus atributos e desafio(s). O dispositivo aguarda o retorno de sua requisição após a convergência de confiança feita pelo domínio e coordenada pelo gerenciador;

Restrito (membro novo): Este estado consiste no acesso limitado ao domínio. O domínio irá acompanhar se aquele dispositivo é confiável (conforme esperado na sua operação). Em toda operação do dispositivo, seus pares vão contabilizando o fator social (MS). O gerenciador continua contabilizando fator de risco e confiança deste dispositivo;

Membro (pleno): Este é o estado em operação completa, o dispositivo está ativo e com capacidade máxima de contribuição no domínio (inclusive auxiliando a classificar novos dispositivos e operações delegadas pelo gerenciador).

Em cada um dos estados listados, há transições que são realizadas conforme a aplicação da política e limiares atingidos. Caso o dispositivo esteja em operação completa, pode receber (baseado na capacidade computacional) do gerenciador tarefas auxiliares como participar de eleição, definição de pares de dispositivos que participarão em desafios, entre outras. Nesse mesmo estado, caso o dispositivo tenha uma pontuação de risco alta (que ultrapasse o limiar configurado), pode ser rebaixado ao estado de operação limitada. Caso, durante a operação limitada, o dispositivo falhe em responder desafios e não tenha uma pontuação alta, pode ser novamente rebaixado a um estado pendente ou até mesmo para fora do domínio (voltando para o estado inicial).

Todas as transições são baseadas em eventos orquestrados pelo gerenciador. Vistos os estados, tem-se as transições definidas (com o formato: *estado anterior* $\rightarrow$ *posterior*):

1. *Send Auth-Req with Attributes* $\triangleright(1) \rightarrow (2)$: Envia o pedido de autorização com atributos;
2. *Access Forbidden* $\triangleright(2) \rightarrow (1)$: Acesso negado;

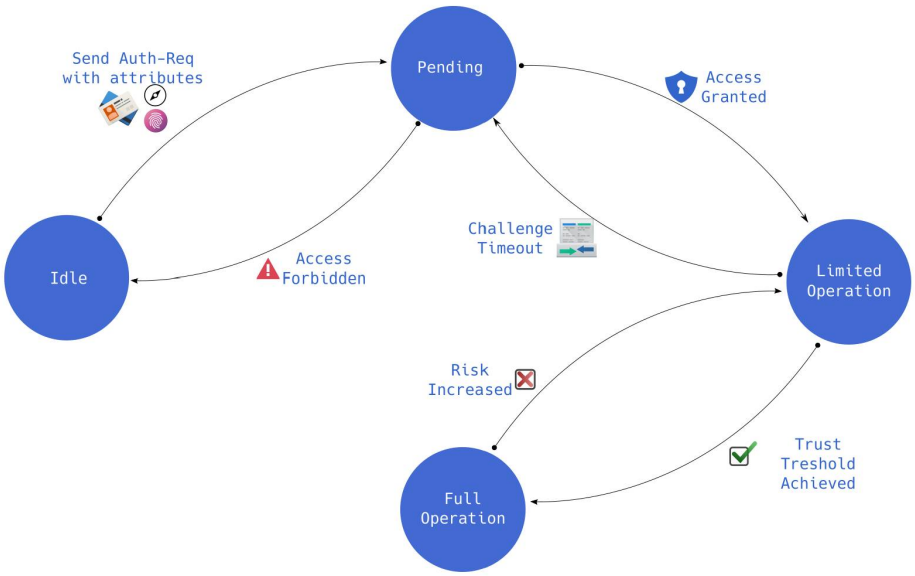


Figura 4.1: Diagrama de estado: ciclo de vida em operação.

3. *Access Granted* ▷(2) → (3): Acesso permitido;
4. *Challenge Timeout* ▷(3) → (2): Desafios expirados;
5. *Trust Treshold Achieved* ▷(3) → (4): Limiar de confiança mínima atingido;
6. *Risk Increased* ▷(4) → (3): Limiar de risco ultrapassado.

4.4 FASES DO PROTOCOLO

As etapas de funcionamento do ciclo de vida do protocolo estão descritas nas fases a seguir. Cabe frisar que há uma fase especial, que ocorrerá somente durante o surgimento de um novo domínio: Criação. Após esta, os dispositivos irão transitar entre a inicialização, operação e eleição. A saída do domínio pode ser realizada de forma voluntária ou por *timeout*.

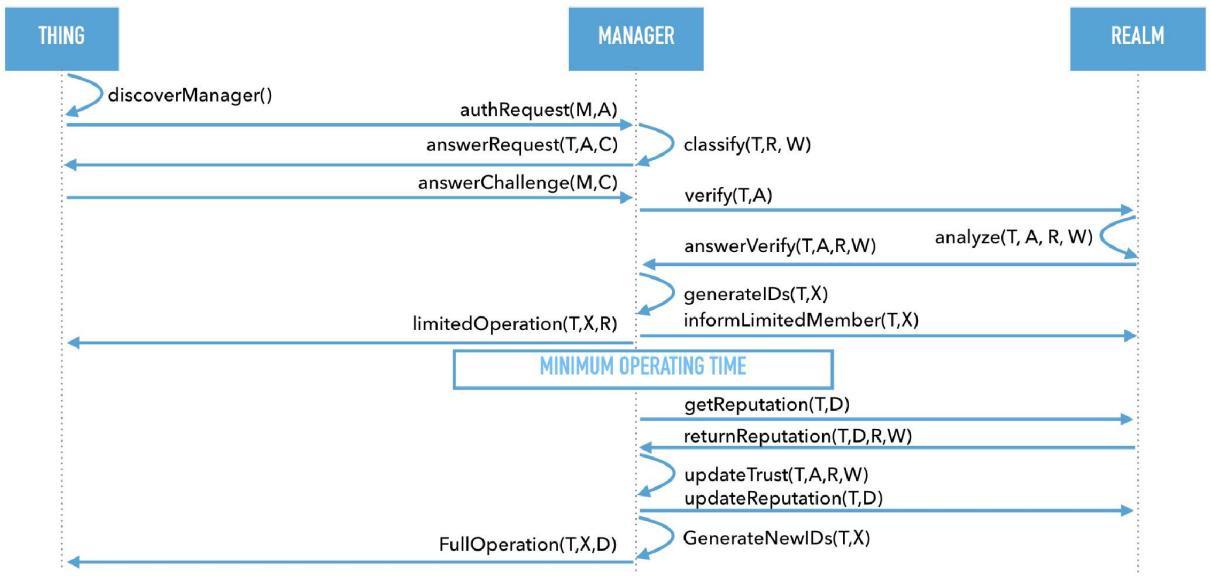


Figura 4.2: Diagrama de sequência.

Em todas as etapas, usar-se-á as identificações: Gerenciador (M), Dispositivo (T), Domínio (D), Atributos (A), Risco (R), Confiança (W), Desafio (C) e Certificados (X). Quando for tratado sobre troca de chaves e a comunicação entre dois dispositivos (podendo ser um dispositivo qualquer e o gerenciador, por exemplo), será tratado como A e B ("Alice & Bob").

4.4.1 Criação

A primeira etapa do protocolo é o caso em que um dispositivo realiza o processo de descoberta e não há nenhum domínio ativo. Portanto, este cria um domínio novo (após confirmação do usuário) e torna-se o gerenciador deste. O dispositivo seguinte irá realizar a troca de chaves similar ao *secure simple pairing*[1] (SSP). Esta versão é apresentada e discutida em [63].

De forma simplificada, o SSP funciona da seguinte maneira: assume-se que há um dispositivo A que inicia o pareamento. Primeiro, A e B trocam chaves públicas PK_a e PK_b - estas chaves são geradas através de curvas elípticas + *Diffie-Hellman*. Depois A e B geram valores randômicos n_a e n_b, respectivamente. B calcula o valor c_b, o qual é uma função resistente à colisão de ambos valores (*Diffie-Hellman* e n_b), e envia-o (c_b) para A. A manda n_a para B. B manda seu *nonce* n_b para A. A recalcula o valor c_b e verifica se este é igual ao valor previamente recebido de B. Note-se que até o momento não houve autenticação - portanto um atacante poderia estar controlando o meio de comunicação [63].

Para verificar se há o "casamento" do pareamento, cada dispositivo calcula um *hash* $H(PK_a, PK_b, n_a, n_b)$ das chaves públicas e ambos *nonces*. O valor do *hash* é truncado para 6 dígitos decimais, os quais são apresentados em ambas as telas. É pedido ao usuário a confirmação da igualdade dos números apresentados. Caso positivo, os dispositivos são considerados *autenticados*, e a chave simétrica K_{init} é derivada dos valores dos *nonces* + *Diffie-Hellman* normalmente [63].

Após esta troca de chaves, os dois dispositivos realizam a eleição do gerenciador do domínio (a eleição é descrita em 4.4.4). Cabe frisar que o usuário escolherá qual uso do domínio, fazendo que o gerenciador crie o mesmo e inicie a operação. Ao escolher este uso, o gerenciador pode sugerir nomes adequados, porém ficando a cargo de cada implementação.

4.4.2 Inicialização

Um dispositivo, ao conectar-se a rede, realiza a descoberta para listar os domínios presentes. Após a confirmação do domínio a participar, este realiza a sequencia apresentada no diagrama visto na figura 4.2.

Cada função, tem a seguintes premissas e objetivos:

- `discoverManager()`: Esta função realiza o *multicast* na rede para identificar se há domínio(s) presentes. Aqui é realizado o *broadcast* de domínio, pois o dispositivo vai listar quais domínios existem e o usuário decidirá qual este participará[2];

- `authRequest()`: Nesta o dispositivo entrante no domínio envia ao gerenciador M, os seus atributos A. Inclui-se, uma sub-rotina de autenticação mútua, para envio e recebimento de mensagens de forma segura;

[1] O SSP é um protocolo de trocas de chaves que procura simplificar o processo de "pareamento"entre dispositivos. O mesmo está disponível na especificação do *Bluetooth*, versão 2.1 + EDR (ou superiores).

[2] Aqui cabe frisar que a interação com o usuário se faz conforme a capacidade do dispositivo para tal. Limitando-se e habilitando "autoconfigurações" conforme a referida capacidade.

- `classify()`: O gerenciador classificará o dispositivo T baseado nos atributos (ambos primários e secundários). Para tal tarefa, levará em conta as saídas da classificação do risco R e confiança W, agregadas à política adotada. Com isso irá criar um desafio C para enviar ao dispositivo;

- `answerRequest()`: O gerenciador, após classificar o dispositivo, cria um desafio baseado nos atributos do mesmo e envia para T;

- `answerChallenge()`: O dispositivo responde o(s) desafio(s) ao gerenciador conforme requisitado. Aqui cabe frisar que o gerenciador pode requerer a resposta de um ou mais desafios, dada a política ou classificação de risco;

- `verify()`: Após receber a resposta do(s) desafio(s), o gerenciador irá realizar a verificação com o domínio, de acordo com o desafio pedido. Por exemplo: se um desafio consiste no dispositivo medir a temperatura do ambiente, o gerenciador faz a votação com outros dispositivos do domínio e verifica se o dispositivo entrante está correto (no mesmo ambiente). Cabe ressaltar que o gerenciador sempre irá avaliar quais sensores e possibilidades de desafio/resposta são factíveis no domínio, para evitar *deadlocks*;

- `analyze()`: Cada membro do domínio que receber uma tarefa de verificação, irá responder conforme suas capacidades. Realizando votação nos casos em que haja mais de um tipo de sensor no domínio;

- `answerVerify()`: Os membros do domínio respondem as respectivas leituras do ambiente;

- `generateIDs()`: Dadas as opções de privacidade do domínio, o gerenciador pode gerar um ou mais identificações, e consequentemente os certificados X, para o dispositivo. Cada dispositivo poderá ter de 1 a $T_A^{1..A}$, onde pode-se ter desde somente uma identidade ou a combinação de cada sensor/atributo que o dispositivo possuir;

- `informLimitedMember()`: O gerenciador comunica os outros membros do domínio em que há novo(s) dispositivo(s), porém ainda de forma limitada;

- `limitedOperation()`: O gerenciador encaminha pro dispositivo, uma ou mais identificações e respectiva(s) chave(s).

Após esta sequência, o dispositivo estará operando de forma limitada, porém já é possível ter interações com outros dispositivos e começar a gerar confiança.

4.4.3 Operação

Nesta etapa realiza-se a gestão de confiança entre os dispositivos. Após a inicialização, de tempo S (definido na política), o gerenciador realizar uma análise de reputação do(s) dispositivo(s) que estão operando de forma limitada. O gerenciador classifica-os, distribui essa informação para os membros plenos do domínio e promove/rebaixa dispositivos conforme pontuação. Observado na figura 4.2:

- `getReputation()`: Esta função é chamada em intervalos de tempo pré-definidos. O gerenciador envia pra todo os membros do domínio o grau de confiança em relação do dispositivo T;

- `returnReputation()`: O gerenciador recebe as respostas dos membros do domínio para consolidação;
- `updateTrust()`: O gerenciador atualiza a tabela de confiança do domínio conforme a resposta anterior e políticas definidas;
- `updateReputation()`: O gerenciador envia a atualização da tabela de confiança pra todos os membros;
- `generateNewIDs()`: Caso haja algum dispositivo que esteja em operação limitada e atingiu o nível de confiança mínimo para operação completa, o gerenciador gera novos IDs e envia para os respectivos dispositivos;
- `fullOperation()`: Ao receber a comunicação do gerenciador, o dispositivo está apto a participar em todas as tarefas do domínio, incluindo sobre a confiança de todos os membros.

Cabe ressaltar que a privacidade ocorre naturalmente, pois cada dispositivo pode criar uma ou mais identidades (conforme política do Gerenciador). Conforme já visto, além da possibilidade de múltiplas identidades por dispositivo, há também a CA do gerenciador, gerando certificados para realização de comunicações seguras ponto-a-ponto.

A única mensagem que o dispositivo *A* receberá do gerenciador, nesta etapa, será a mensagem que encaminha para *A* os novos certificados (com maior tempo de validade), além da matriz de risco do domínio. Finalizando a entrada do dispositivo *A* no domínio e este operando de forma completa.

4.4.4 Eleição

Como IoT é um ambiente dinâmico, há a possibilidade do gerenciador ficar indisponível, seja por deslocamento (fora do alcance da rede) ou por falta de energia. Se faz necessária uma nova eleição do gerenciador pelos membros do domínio. A eleição sempre priorizará o dispositivo com maior poder computacional, pois a sobrecarga do domínio é de responsabilidade do gerenciador. O mecanismo da eleição funciona da seguinte forma:

1. Após o fim do intervalo de tempo de espera de resposta do gerenciador, propriedade do domínio. Cada membro envia para outros membros do domínio um desafio para prova de trabalho (*proof of work* - PoW);
2. Cada membro gera um número aleatório único (*nonce*) com respectivo horário e envia para cada membro do domínio;
3. Cada membro, ao receber o número e o ID da origem, este calcula um PoW e retorna para o respectivo membro;
4. Ao receber o membro ordena a tabela conforme a capacidade computacional dos outros pares;
5. O consenso do dispositivo que contém o maior poder computacional é eleito o gerenciador do ambiente.

Cabe ressaltar que na eleição só participam os dispositivos que estão em operação completa. Durante a transição os dispositivos em operação limitada aguardam a convergência da votação. Uma forma de implementação que pode minimizar a necessidade de eleição seria ter a "lista tripla" dos maiores poderes computacionais do domínio.

Outro ponto importante do comportamento, é no caso do novo dispositivo em operação completa. Este ao ser notificado da mudança de estado, irá realizar um PoW para avaliar o poder computacional e saber qual ponto da hierarquia estará.

4.4.5 Considerações Finais

Neste capítulo apresentou-se o modelo de confiança e as fases do protocolo. Este modelo baseia-se nos desafios para gerar um coeficiente de classificação de risco de cada dispositivo. A seguir, no Capítulo 5, serão avaliados o modelo de ameaças, respectivas mitigações, detalhes de implementação e observação e uso dos atributos.

5 AVALIAÇÃO

Este capítulo apresenta um resumo das avaliações pertinentes ao protocolo e os detalhes de implementação da proposta e testes. Além deste detalhamento, apresenta-se uma avaliação e discussão do modelo de ameaças e respectivas mitigações. Também é apresentado uma descrição dos experimentos e os resultados obtidos com estes.

5.1 IMPLEMENTAÇÃO

Foram realizados protótipos como prova de conceito (PoC) para testes iniciais de validação do uso dos atributos pelo protocolo. Esta PoC foi codificada em *Python*, versão 3.7 e respectivas dependências (ver anexo A). Nestes testes o protocolo foi implementado utilizando como base de comunicação o MQTT. A implementação leva em conta a possibilidade de diferentes formas de transporte dos dados (que no protótipo foi realizado pelo MQTT), passível de extensão e melhorias. A escolha do MQTT se faz por ser um padrão emergente em aplicações de IoT. Vários trabalhos dão ênfase no uso de MQTT - sem esquecer de ZeroMQ, AMQP e XMPP [64, 40].

Na Figura 5.1, apresenta-se o diagrama das classes implementadas dos testes realizados (a versão expandida está presente no Anexo A):

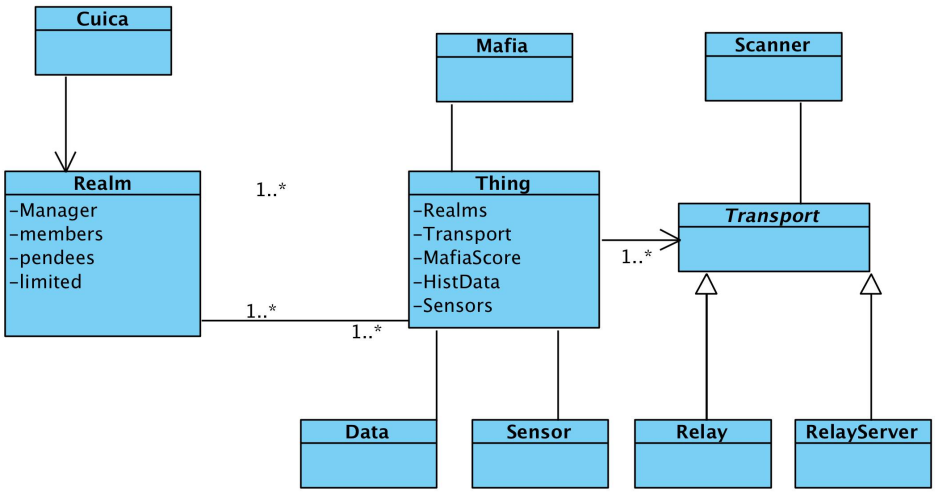


Figura 5.1: Diagrama Diagrama de Classes.

Os códigos executaram em três *Raspberry Pi* com circuitos contendo fotorresistores para mensuração da luz no ambiente - conforme visto no Anexo A. Estes dispositivos executavam o sistema operacional *Raspbian GNU/Linux*, versão 9.

Alguns cenários de uso do protocolo com os respectivos contextos podem auxiliar na aplicação e uso em soluções:

- **Casa**: cenário base apresentado durante o trabalho, o uso do contexto em um ambiente doméstico. Tem-se o uso de luminância bem distinto entre cômodos, há variação de sons em cada ambiente, e em alguns casos, há até variação de temperatura (cômodos que tem ar-condicionado e o que não têm). A possibilidade de uso de câmera também é possível.

- **Comércio**: o cenário de uso comercial (escritórios), pode haver pouca efetividade na luminância e temperatura, já que tem a característica de uma concentração maior de pessoas e horários de uso mais uniforme - não gerando diferenciação suficiente nestas duas observações. Por outro lado, há a diferença maior no som entre ambientes e há também a maior possibilidade de presença de câmeras, auxiliando nos desafios de câmera/LEDs.

- **Universidades/Escolas**: cenário similar ao comercial, com probabilidade do som ambiente ser maior diferencial entre salas de aulas e outros ambientes. Luminância pode ser um apoio interessante - aulas com projeção, etc..

- **Agricultura**: cenário interessante de uso, onde som não deve auxiliar e geralmente não deve haver câmeras. Restando luminância, temperatura e umidade como apoio e caracterizado localidade e proximidade.

- **Indústria**: cenário onde todos as observações devem apresentar uma diferenciação grande entre ambientes.

5.2 AVALIAÇÃO DA SEGURANÇA E MODELO DE AMEAÇAS

Protocolos consistem em ações sequenciais entre duas ou mais entidades de forma a atingir um objetivo. Este objetivo, na segurança, pode ser: autenticação, autorização ou distribuição de chaves, para exemplificar alguns. Há de se reconhecer que existem distintas necessidades e especificações para objetivos de autenticação válidos. Situações diferentes podem requerer algoritmos de criptografia mais fraca ou mais forte, dependendo do uso. Desta forma, faz-se necessário que protocolos de uso comum tenham a premissa de adaptar-se a distintos usos [63].

Agregando as premissas de Needham-Schroeder às premissas de Dolev-Yao, tem-se um modelo consensual, aplicável pela comunidade de segurança. Consistem nestas premissas: criptografia infalível; atacante que controla os canais de comunicação; membros observantes do protocolo; atacante capaz de particionar mensagens até seus componentes básicos; atacante capaz decifrar as mensagens de que dispõe as chaves; atacante capaz reencaminhar mensagens criptografadas; e por fim, o atacante que é, também, um agente interno do protocolo, executando-o para conseguir informações adicionais a seu favor [63].

Neste trabalho, será aplicado o modelo de ameaças da família BUG. Este modelo divide os participantes do protocolo em grupos: Mau, Feio e Bom - especificamente a variante *Multi-Attacker*. A família BUG é importante pois tem premissas de que há vários atacantes e estes não compartilham informações, além de mudarem de comportamento durante a execução do protocolo (dependendo do custo/benefício do ataque). A variante *Multi-Attacker*, tem a premissa de que o atacante nunca irá compartilhar seus segredos de longo prazo, pensando somente no ganho próprio [63].

5.2.1 Cenários de Ataques e Mitigações ao Protocolo

Os principais cenários de ataques estão listados a seguir, junto com as respectivas discussões sobre cada caso:

Dispositivo malicioso: Um dispositivo malicioso tenta ingressar ao domínio. Existem dois sub-casos: este dispositivo encontra-se no mesmo ambiente ou está remoto. No primeiro, este dispositivo observará o protocolo e quando iniciar as ações maliciosas, irá começar a

afetar o próprio MS, sendo demovido e eventualmente banido do domínio. No segundo, estando remoto, o dispositivo não responderá de maneira rápida e correta os desafios, não conseguindo participar do domínio;

Poder computacional: Seria outro caso específico de um dispositivo malicioso, que detém a característica de muito poder computacional. Neste caso, este deve ter a capacidade maior que todos os outros dispositivos do domínio, além de estar no mesmo ambiente. Neste caso é possível ganhar o papel de gerenciador. Até o passo final - ganhar o domínio - o dispositivo terá que se comportar conforme o protocolo. Porém, ao se tornar o gerenciador, este traz os holofotes para si, interagindo com o usuário e explicitando que há um dispositivo não conhecido, gerando questionamentos. Dois contra-incentivos neste caso: custo e exposição. O atacante necessitará de investimento para colocar o dispositivo e terá atenção do usuário;

Poder de votação distribuído: Vários dispositivos maliciosos ingressam e tentam controlar a votação. Seria uma variante do caso anterior, porém agora com vários dispositivos tentando influenciar tanto a votação do gerenciador, e/ou até mesmo a classificação de riscos e leituras de informações do ambiente (tornando dispositivos válidos em "maliciosos"). Aqui o custo e os holofotes também tornam-se contra-incentivos;

Bloqueio de Comunicação: Um dispositivo malicioso tenta burlar desafios, bloqueando comunicação entre outros dispositivos do domínio. Neste caso, o gerenciador não irá promover os dispositivos, trazendo holofotes para problemas que seriam aparentemente da infra-estrutura de comunicação.

Cabe frisar que o dispositivo malicioso que estiver desde o início (etapa de criação) no domínio poderá participar e se tornar um membro pleno do mesmo. O contra-incentivo existente é ter que existir o dispositivo no domínio desde a criação. Outro caso é no dispositivo que apresente comportamento malicioso de forma esporádica, não haverá impacto grande no seu coeficiente e, portanto, podendo se tornar um membro pleno e consistir em um ataque bem sucedido. Neste caso, necessita-se de ajustes finos e dados mais precisos para conseguir identificar estes comportamentos e sinalizar o dispositivo malicioso de forma satisfatória - ficando como parte de trabalhos futuros.

A resiliência do protocolo é suficiente para o dia-a-dia. Tem-se um equilíbrio entre praticidade e segurança, tornando custoso e explícito dispositivos maliciosos que atinjam sucesso nos ataques.

5.3 ANÁLISE DO USO DE ATRIBUTOS

Os primeiros testes visaram analisar a variação da luminância em um mesmo cômodo (em pontos distintos) de uma residência - apresentados na Figura 5.2 . Incluiu-se um dispositivo com 4 sensores, para avaliar se haveria diferença na variação da leitura entre sensores de um mesmo ponto. As correlações foram calculadas usando coeficiente de *Pearson*.

No mesmo dispositivo, Figura 5.2, em sensores distintos, houve uma correlação de $0,977 \pm 0,007$ - portanto observa-se que não houve problemas entre sensores. Tem-se, portanto, uma confiabilidade nas leituras, caso estes sejam separados ou recombinados. Já entre dispositivos distintos (continuando no mesmo cômodo), a correlação foi de $0,864$ - calculada entre a média do primeiro (com 4 sensores) com as leituras do segundo. Todas as leituras foram normalizadas antes do cálculo de correlação. Para entender melhor gráfico, deve-se levar em consideração que

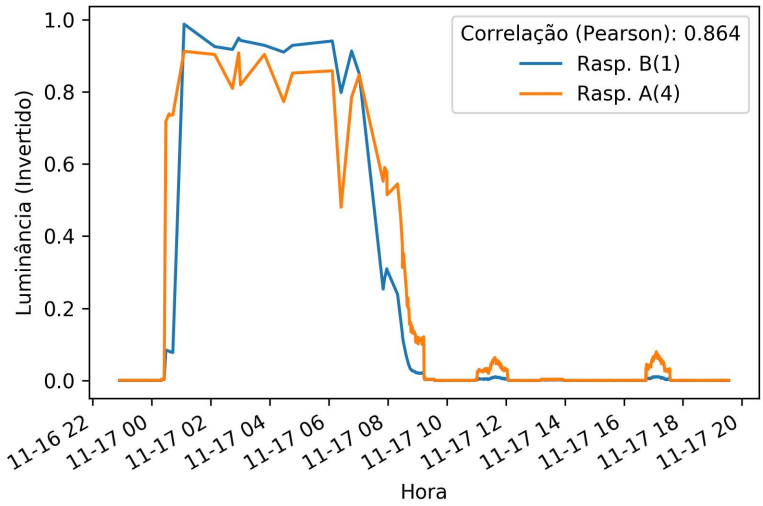


Figura 5.2: Variação de luminância no mesmo cômodo.

a leitura é inversamente à quantidade de luz do ambiente. Portanto, quanto menor a quantidade de luz, menor será a resistência e maior será o valor lido pela porta.

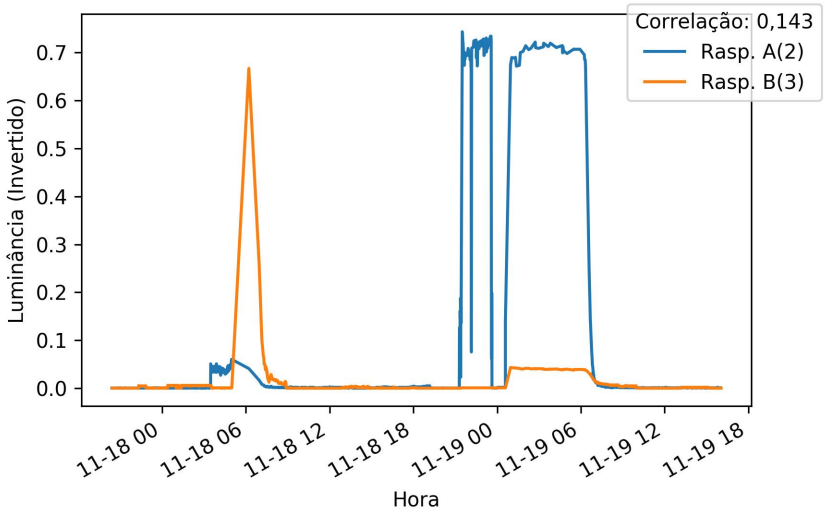


Figura 5.3: Variação de luminância em cômodos distintos.

Nos testes entre cômodos distintos, Figura 5.3, foram distribuídos os sensores novamente. No caso do dispositivo com dois sensores, a intra-correlação foi de $0,979$, já o dispositivo com três sensores a intra-correlação foi de $0,991 \pm 0,003$. A correlação entre dispositivos foi de $0,143$. Nota-se, com isso que a luminância auxiliou uma identificação de não-correlação entre os dados observados em cada dispositivo - estando estes em cômodos distintos.

Há de se considerar que um ambiente doméstico é mais propício a variação de luminância entre cômodos, pois os hábitos de pessoas distintas tem uma influência grande no cômodo.

5.3.1 Uso de Média Móvel

Este trabalho levantou questões complementares às dúvidas iniciais. Algumas possibilidades mostram-se válidas para continuidade de pesquisa: realizar as observações com intervalo de tempo maiores, avaliação de outros tipos de sensores, variações de combinação entre

sensores, variação no histórico social, escalabilidade. Além de outros cenários distintos de uso e respectivos impactos para avaliação: universidades, comércios, indústrias, cidades, são alguns exemplos pendentes de análise.

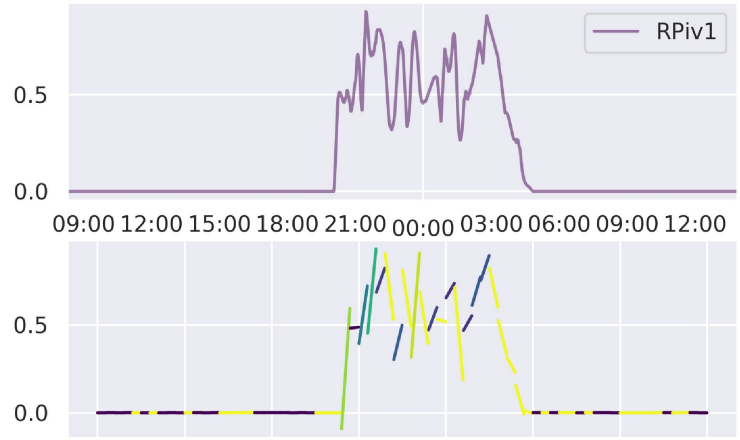


Figura 5.4: Variação do coeficiente dos segmentos de reta (Dispositivo 1).

Outra opção de testes consiste em avaliar uma possibilidade advinda ao final do desenvolvimento deste: a avaliação do conceito de média móvel com os dados de luminância. A depender das características de poder computacional limitado, qual seria a eficácia - e o respectivo custo computacional - para validação do comportamento dos dados de luminância, através de regressão linear?

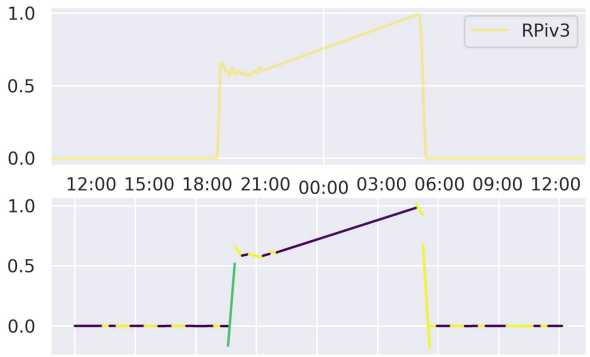


Figura 5.5: Variação do coeficiente dos segmentos de reta (Dispositivo 2).

Conforme observado nas Figuras 5.4 e 5.5, nota-se que o comportamento do dado observado apresenta sua tendência. Nos gráficos superiores, tem-se a observação do dispositivo, com a média entre os 3 fotorresistores. Nos gráficos inferiores, tem-se segmentados, os conjuntos de observações e respectivas inclinações de reta - conforme variação de cores. No caso de inclinações negativas, estão denotadas em tons de amarelo, os casos de inclinações positivas, nas cores restantes. Assim, na Figura 5.4 é apresentado o gráfico do dispositivo #1 e na Figura 5.5 é apresentado o gráfico do dispositivo #3., apresentados como exemplos.

As Figuras 5.6 e 5.7 mostram análises realizadas do uso do inclinação da reta (baseada nas observações tomadas nos últimos 10 minutos). A Figura 5.6 apresenta o gráfico superior a média de leitura dos 3 fotoresistores, em cada dispositivo. No gráfico inferior estão os coeficientes de reta calculados de cada dispositivo (conforme amostragem e média das leituras

Figura 5.6: Observações no mesmo Cômodo.

do fotoresistores). No caso da Figura 5.7 (com a mesma disposição da figura anterior), tem-se uma correlação discrepante do RPi n^o3 - conforme esperado.

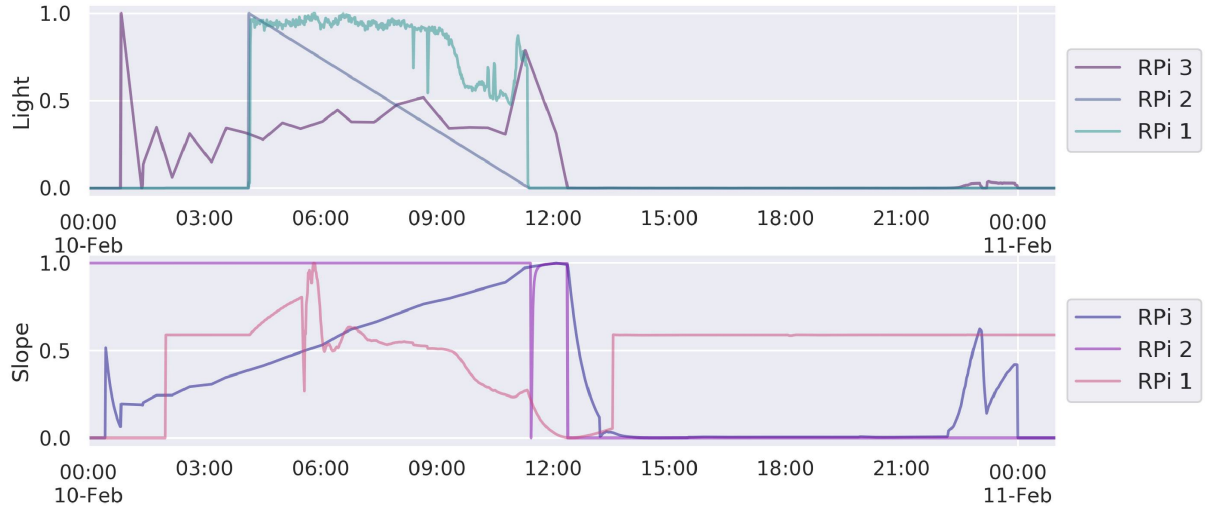
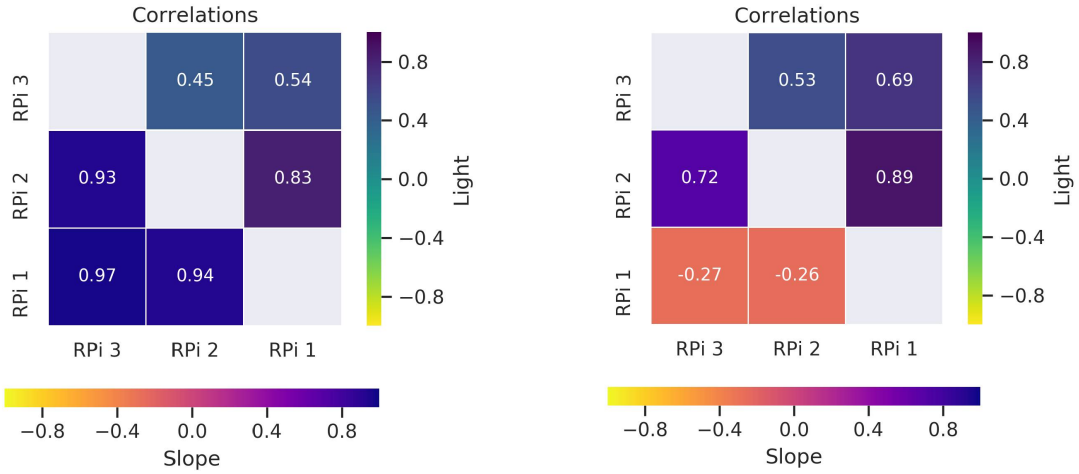


Figura 5.7: RPi n^o3 em outro Cômodo.

Nas Figuras 5.8(a) e 5.8(b) estão as duas matrizes de correlação combinadas: o triângulo inferior encontram-se as correlações das inclinações das retas e no triângulo superior as correlações dos dados observados. Observando a Figura 5.8(a), conclui-se, de acordo com a matriz, que a inclinação da reta apresentou uma correlação maior que os dados brutos - todos os dispositivos observavam o mesmo ambiente.

No caso da Figura 5.8(b), o dispositivo n^o3 é discrepante - conforme apresentada na correlação dos dados observados (triângulo superior). Porém na correlação do coeficiente de reta, apresentou duas anomalias entre as relações dos dispositivos. Observou-se na relação n^o3 e n^o1 uma correlação alta (esperava-se coeficiente baixo); no caso n^o2 e n^o1 uma correlação baixa (esperava-se coeficiente alto). Estas surpresas indicam algumas possibilidades: melhorar a escolha da taxa de amostragem, melhorar os ajustes nos dados para remoção de pontos discrepantes ou com erro, aumentar tempo de observação de cômodos e entre cômodos. Fica em



(a) Correlações no mesmo cômodo.

(b) Correlações em cômodos distintos.

Figura 5.8: Variação das correlações entre os coeficientes de reta (inferior) e dados observados (superior).

aberto a validação do uso de memória e processamento em dispositivos limitados - os quais não foram mensurados durantes os testes.

5.4 CONSIDERAÇÕES FINAIS

Este Capítulo apresentou as principais avaliações sobre o protocolo. Apresentou-se as classes implementadas (com maiores informações no Anexo A). Discutiu-se o modelo de ameaças e a respectiva resiliência do protocolo. A avaliação do uso de atributos apresentou as características interessantes do uso da luminância para verificar o pertencimento dos dispositivos no mesmo ambiente. Inclui-se a possibilidade de uso de uma média móvel também pode ser útil para minimizar um grande número de pontos de histórico, diminuindo a quantidade de informação armazenada e uso de memória.

6 CONCLUSÕES

A massificação da Internet das Coisas acarretou novos desafios de segurança. Dispositivos inseguros podem vazar dados - desde a senha do *Wi-Fi* doméstico a dados de agentes de segurança, incluindo assassinos de aluguel[1]. Pesquisar protocolos de autorização que facilitem o seu uso continua necessário. Este trabalho apresentou uma proposta de protocolo de autorização para Internet das Coisas baseado em atributos e informações de contexto.

No que tange ao modelo de ameaças, o protocolo resiste satisfatoriamente. O custo do ataque torna-se desinteressante - com ressalvas a agentes de estado. O uso de informações de ambiente se mostra promissor como suporte à segurança e facilidade de uso. A luminância pode ser usada, entre dispositivos, como fonte indicativa de pertencimento ao mesmo ambiente. Cabe ressaltar que, mesmo com indicativo promissor de uso, existem variáveis que podem auxiliar ou atrapalhar esta observação específica: a localização intra-cômodo, quantidade de pessoas e tipo de uso do cômodo. No cômodo, os dispositivos podem estar: cobertos por sombra, próximos de fontes de luz adversas à iluminação do ambiente, próximos a janelas, etc. Portanto, a luminância pode ser uma observação de grande valia, mas não como elemento único de observação - ao contrário de outras abordagens, tais como áudio [65]. Carece de maiores testes e mais dados para comparar quais elementos de observação são suficientes ou necessitam de outras observações complementares. Combinados, vários atributos podem melhorar a contextualização e segurança do domínio.

Cabe ressaltar a necessidade de aprofundar a análise formal da gestão do domínio de confiança - através da formalização da especificação do protocolo. Modelar o protocolo para avaliação em ferramentas de análise simbólica, como Scyther ou Avispa, é uma tarefa em aberto. Além da formalização, a viabilidade ficou carente de testes e validação de escalabilidade - sem a possibilidade de trazer respostas mais consistentes. Por fim persiste validar a eficiência do protocolo em dispositivos mais restritos computacionalmente - como *Arduino/Microduino*, `ESP32`, por exemplo.

Apresentou-se também um compilado sobre o tema de autorização para IoT, bem como sobre os principais trabalhos da atualidade. Observou-se a falta de padrões de mercado que levem em consideração a dificuldade da população com a tecnologia, que geralmente acarreta em decisões inseguras. Há poucos trabalhos que apresentam uso de dados de contexto como apoio às decisões de segurança - destacando-se o uso de áudio, já comentado. Os testes realizados mostraram, de forma promissora, a viabilidade do uso da luminância como fator auxiliar de autorização. Além disso, foi analisado o modelo de ameaças a que o protocolo resiste satisfatoriamente para uso doméstico.

Dada a hipótese de pesquisa, este trabalho demonstrou resultados inicias sobre uso de informações de contexto como suporte para a autorização é promissor. Usar luminância mostrou-se promissor, demonstrando pertencimento conforme os testes de localidade de cômodos. Porém o uso da luminância e as outras informações de contexto demanda continuidade de pesquisa e um maior volume de dados para o uso ser conclusivo. Continua em aberto a necessidade de validar se o uso dos dados do ambiente facilita-se o uso (por mais que alguns projetos já usam áudio, por exemplo). Além disso, houve a contribuição do código. Todo código gerado está disponível como base para continuidade e extensão deste trabalho[2]

[1] Notícia: https://goo.gl/PQMLMd

[2] O código encontra-se em https://gitlab.c3sl.ufpr.br/jrquerubin/cuica.

Entretanto, alguns questionamentos que ainda persistem, abrindo a oportunidade de novas possibilidades de pesquisas futuras. Recapitulando, os trabalhos futuros listados: avaliação formal de ambos, protocolo e gestão de confiança entre dispositivos; viabilidade de uso do protocolo em dispositivos mais restritivos; escalabilidade; e, eficiência. Destaca-se a falta de trabalhos que façam um comparativo mais amplo entre as informações de contexto. As observações físicas e respectivas combinações, fecham as possibilidades futuras de pesquisa em segurança para IoT apresentadas neste trabalho.

REFERÊNCIAS

[1] ITU-T. The tactile internet. Technical report, ITU-T, August 2014.

[2] M.A. Monteiro. *Introdução à organização de computadores*. Livros Técnicos e Científicos, 2002.

[3] William Stallings. *Computer Organization and Architecture: Designing for Performance (8th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2010.

[4] Kalle Lyytinen and Youngjin Yoo. Ubiquitous computing. *Communications of the ACM*, 45(12):63–96, 2002.

[5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE CST*, 16(1):414–454, First 2014.

[6] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[7] Chetan Sharma. Correcting the IoT history. https://goo.gl/HnwYrs, 2016. Acessado em 26/11/2016.

[8] Rolf H. Weber. Internet of things – new security and privacy challenges. *Computer Law & Security Review*, 26(1):23 – 30, 2010.

[9] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497 – 1516, 2012.

[10] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645 – 1660, 2013.

[11] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things: perspectives and challenges. *Wireless Networks*, 20(8):2481–2501, 2014.

[12] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2360–2367. IEEE, 2010.

[13] Jules Polonetsky, Omer Tene, and Kelsey Finch. Shades of Gray: Seeing the Full Spectrum of Practical Data De-identification. *Santa Clara Law Review*, 56:594–629, 2016.

[14] R. Roman, P. Najera, and J. Lopez. Securing the internet of things. *Computer*, 44(9):51–58, Sept 2011.

[15] Alex Hern. Fitness tracking app strava gives away location of secret us army bases. https://goo.gl/E84zFU, 2018. Acessado em 29/01/2018.

[16] Matt Burgess. Strava's data lets anyone see the names (and heart rates) of people exercising on military bases. https://goo.gl/wvk7QQ, 2018. Acessado em 30/01/2018.

[17] Nora Young. Exercise app shows why anonymous data can still be dangerous. https://goo.gl/vX4SzG, 2018. Acessado em 02/02/2018.

[18] Jo Ann Oravec. Emerging "cyber hygiene"practices for the IoT. In *2017 IEEE International ProComm*. IEEE, 2017.

[19] Patricia Arias-Cabarcos, Florina Almenarez, Ruben Trapero, Daniel Diaz-Sanchez, and Andres Marin. Blended identity: Pervasive IdM for continuous authentication. *IEEE Security & Privacy*, 13(3):32–39, 2015.

[20] Neil Zhenqiang Gong, Altay Ozen, Yu Wu, Xiaoyu Cao, Richard Shin, Dawn Song, Hongxia Jin, and Xuan Bao. Piano: Proximity-based user authentication on voice-powered internet-of-things devices. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 2212–2219. IEEE, 2017.

[21] Hannes Plank, Christian Steger, Thomas Ruprechter, Gerald Holweg, and Norbert Druml. Survey on camera based communication for location-aware secure authentication and communication. *EMC Summit*, April 2016.

[22] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266 – 2279, 2013.

[23] Michelle S Wangham, Marlon Cordeiro Domenech, and Emerson Ribeiro de Mello. Infraestrutura de autenticaçao e de autorizaçao para internet das coisas. In *Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais—SBSeg*, 2013.

[24] Eleonora Borgia. The internet of things vision: Key features, applications and open issues. *Computer Communications*, 54:1 – 31, 2014.

[25] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti, and Subhajit Dutta. *A Survey of Middleware for Internet of Things*, pages 288–296. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[26] Jim Burton. Leverage intelligent gateways in your IoT architecture. https://www.gartner.com/document/code/00301205, 2016. Gartner™.

[27] Google™. Android of things™. https://developer.android.com/things/, 2017. Acessado em 01/01/2017.

[28] Dominique Guinard, Vlad Trifa, and Erik Wilde. A resource oriented architecture for the web of things. In *Proceedings of Internet of Things 2010 International Conference (IoT 2010)*, Tokyo, Japan, November 2010.

[29] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, and Siobhán Clarke. Middleware for internet of things: a survey. *IEEE Internet of Things Journal*, 3(1):70–95, 2016.

[30] M. A. Chaqfeh and N. Mohamed. Challenges in middleware solutions for the internet of things. In *2012 International Conference on Collaboration Technologies and Systems (CTS)*, pages 21–26, May 2012.

[31] T. Pflanzner and A. Kertesz. A survey of IoT cloud providers. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 730–735, May 2016.

[32] Xuezhi Zeng, Saurabh Kumar Garg, Peter Strazdins, Prem Prakash Jayaraman, Dimitrios Georgakopoulos, and Rajiv Ranjan. Iotsim: A simulator for analysing iot applications. *Journal of Systems Architecture*, 72:93 – 107, 2017. Design Automation for Embedded Ubiquitous Computing Systems.

[33] A. Zaslavsky, C. Perera, and D. Georgakopoulos. Sensing as a Service and Big Data. *ArXiv e-prints*, January 2013.

[34] D. Hughes and N. Correll. Distributed Machine Learning in Materials that Couple Sensing, Actuation, Computation and Communication. *ArXiv e-prints*, June 2016.

[35] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys Tutorials*, 16(4):1996–2018, Fourthquarter 2014.

[36] Drue Reeves. Preparing, planning and architecting for the internet of things. https://www.gartner.com/document/code/00292675, 2016. Gartner™.

[37] Paul DeBeasi. Gartner report: Solution path for executing an internet of things initiative. https://www.gartner.com/document/code/00293163, 2016. Gartner™.

[38] Paul Dourish. What we talk about when we talk about context. *Personal and ubiquitous computing*, 8(1):19–30, 2004.

[39] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. *Computer Networks*, 76:146 – 164, 2015.

[40] Omer Berat Sezer, Erdogan Dogdu, and Ahmet Murat Ozbayoglu. Context-aware computing, learning, and big data in internet of things: a survey. *IEEE Internet of Things Journal*, 5(1):1–27, 2018.

[41] European Parliament and Council of the European Union. Directive 2016/679/EU, 2016.

[42] W3C Working Group. The platform for privacy preferences 1.1 (p3p1.1) specification. https://www.w3.org/TR/P3P11/, 2006. Acessado em 19/01/2017.

[43] C. Sengul. Privacy, consent and authorization in IoT. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 319–321, March 2017.

[44] Alan Karp, Harry Haury, and Michael Davis. From abac to zbac: the evolution of access control models. In *International Conference on Cyber Warfare and Security*, page 202. Academic Conferences International Limited, 2010.

[45] DoD Standard. Department of defense trusted computer system evaluation criteria; december 1985, dod 5200.28-std. Technical report, Supersedes CSC-STD-001-83, dtd 15 Aug 83, Library, 1985.

[46] Ferraiolo David and Kuhn Richard. Role-based access controls. In *Proceedings of 15th NIST-NCSC National Computer Security Conference*, volume 563. Baltimore, Maryland: NIST-NCSC, 1992.

[47] R. Shirey. Internet Security Glossary, Version 2. RFC 4949, RFC Editor, August 2007.

[48] Sergio Gusmeroli, Salvatore Piccione, and Domenico Rotondi. IoT access control issues: a capability based approach. In *6th International Conference - IMIS*, pages 787–792. IEEE, 2012.

[49] Ning Ye, Yan Zhu, Ru-chuan Wang, and Qiao-min Lin. An efficient authentication and access control scheme for perception layer of internet of things. *Applied Mathematics and Information Sciences*, 2014.

[50] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, 800(162), 2013.

[51] Dina Hussein, Emmanuel Bertin, and Vincent Frey. A community-driven access control approach in distributed IoT environments. *IEEE Communications Magazine*, 55(3):146–153, 2017.

[52] Aafaf Ouaddah, Hajar Mousannif, Anas Abou Elkalam, and Abdellah Ait Ouahman. Access control in the internet of things: Big challenges and new opportunities. *Computer Networks*, 2017.

[53] Jaehong Park and Ravi Sandhu. The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174, 2004.

[54] Zhang Guoping and Gong Wentao. The research of access control based on UCON in the internet of things. *Journal of Software*, 6(4):724–731, 2011.

[55] Ola Salman, Imad Elhajj, Ali Chehab, and Ayman Kayssi. Software defined IoT security framework. In *2017 Fourth International Conference on SDS*, pages 75–80. IEEE, 2017.

[56] M. Alramadhan and K. Sha. An overview of access control mechanisms for internet of things. In *2017 26th ICCCN*, pages 1–6, July 2017.

[57] John Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo, and Moti Yung. Fourth-factor authentication: Somebody you know. *ACM Trans. Inf Syst. Secur.*, October 2006.

[58] Seungsoo Lee, Changhoon Yoon, and Seungwon Shin. The smaller, the shrewder: A simple malicious application can kill an entire sdn environment. In *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pages 23–28. ACM, 2016.

[59] Zheng Yan, Peng Zhang, and Athanasios V Vasilakos. A survey on trust management for internet of things. *Journal of network and computer applications*, 42:120–134, 2014.

[60] Raquel Lacuesta, Guillermo Palacios-Navarro, Carlos Cetina, Lourdes Peñalver, and Jaime Lloret. Internet of things: where to be is to trust. *EURASIP JWCN*, page 203, 2012.

[61] R McGraw. Risk-adaptable access control (RaDAC). In *Privilege (Access) Management Workshop. NIST – Information Technology Laboratory*, 2009.

[62] Hany F Atlam, Ahmed Alenezi, Robert J Walters, Gary B Wills, and Joshua Daniel. Developing an adaptive risk-based access control model for the IoT. In *IEEE iThings*. IEEE, 2017.

[63] Marcelo Carlomagno Carlos, Jean Everson Martina, Geraint Price, and Ricardo Felipe Custódio. An updated threat model for security ceremonies. In *SAC '13 Conference*. ACM, 2013.

[64] Daniel Happ, Niels Karowski, Thomas Menzel, Vlado Handziski, and Adam Wolisz. Meeting iot platform requirements with open pub/sub solutions. *Annals of Telecommunications*, 72(1):41–52, Feb 2017.

[65] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. Sound-proof: Usable two-factor authentication based on ambient sound. In *USENIX Security Symposium*, 2015.

[66] K. Angrishi. Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets. *ArXiv e-prints*, February 2017.

APÊNDICE A – INFORMAÇÕES DETALHADAS DE IMPLEMENTAÇÃO

Este anexo aprofunda as informações da implementação e testes realizados.

A.1 *HARDWARE*

As informações gerais de cada dispositivo e respectivos circuitos estão listados:

A.1.1 *Raspberry Pi #1*

Este é um *Raspberry Pi* modelo clássico, com:

Informações: Pi 1 Model B Rev 1 (Figura A.1)

CPU: ARMv6-compatible processor rev 7 (v6l). SoC: BCM2835, Revision: 0003;

Memória: 256 MB

Sistema Operacional: Raspbian GNU/Linux 9

Componentes: 3 Fotoresistores, 1 LED RGB (4 pinos), 3 Capacitores de $1\mu F/50V$, 3 resistores de 270Ω e um de $10k\Omega$, um sensor de temperatura DS18B20 e um microfone de ruído.

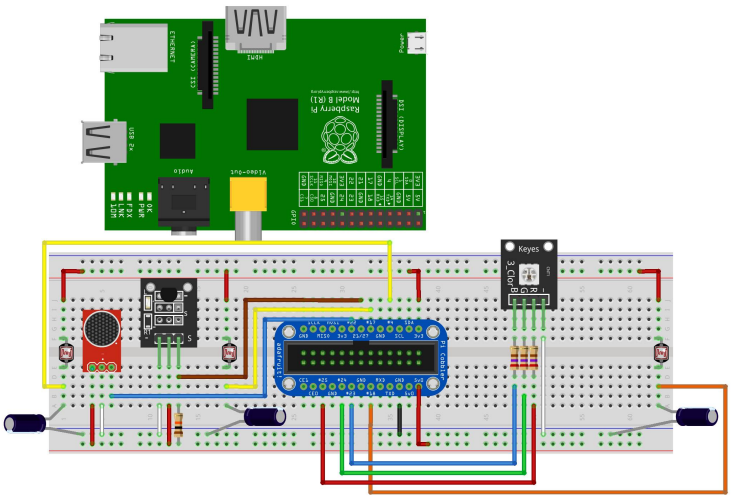


Figura A.1: Diagrama do circuito do RPi #1

A.1.2 *Raspberry Pi #2*

Este é um *Raspberry Pi* modelo clássico, com:

Informações: Pi 1 Model B Rev 1 (Figura A.2)

CPU: ARMv6-compatible processor rev 7 (v6l). SoC: BCM2835, Revision: 0003;

Memória: 256 MB

Sistema Operacional: Raspbian GNU/Linux 9

Componentes: 3 Fotoresistores, 3 Capacitores de $1\mu F/50V$, 3 LED's (R/G/B), 3 resistores de 560Ω e um de 570Ω, um sensor de temperatura/humidade DHT11.

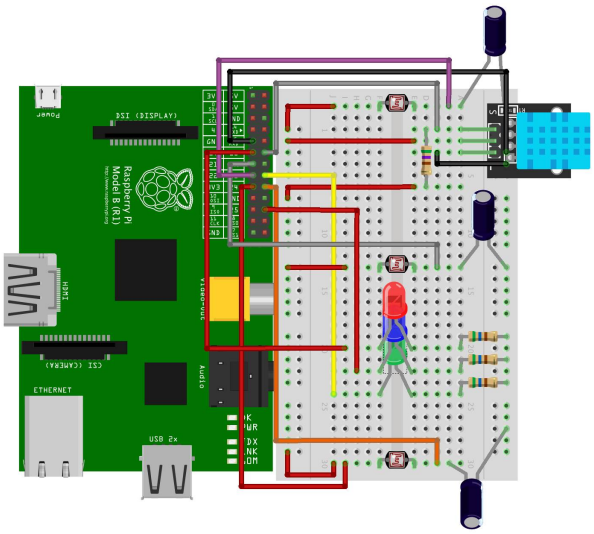


Figura A.2: Diagrama do circuito do RPi #2

A.1.3 *Raspberry Pi #3*

Este é um *Raspberry Pi 3 Model B*, com:

Informações: Pi 3 Model B+ (Figura A.3)

CPU: ARMv7 Processor rev 4 (v7l). SoC: BCM2837, Revision : a020d3

Memória: 1 GB

Sistema Operacional: Raspbian GNU/Linux 9

Componentes: 3 Fotoresistores, 1 LED RGB (4 pinos), 3 Capacitores de $1\mu F/50V$, 3 resistores de 560Ω, um sensor de temperatura/humidade/pressão BMP085 e um *buzzer* ativo.

A.2 *SOFTWARE*

Arquivo de dependências *Python*, `requirements.txt`:

```
pip==18.1
bumpversion==0.5.3
wheel==0.32.1
watchdog==0.9.0
flake8==3.5.0
tox==3.5.2
coverage==4.5.1
Sphinx==1.8.1
```

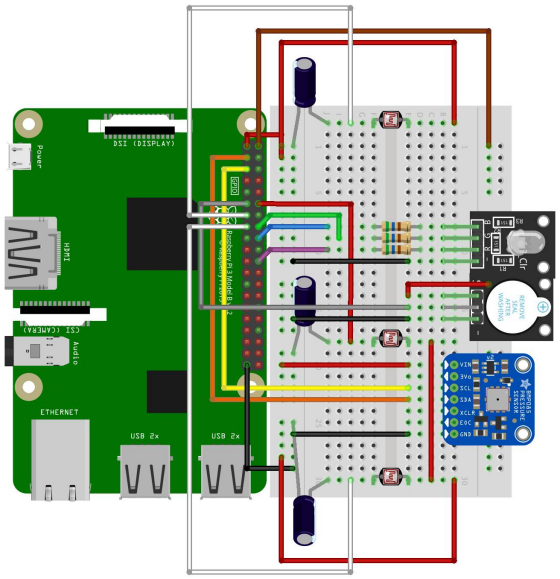


Figura A.3: Diagrama do circuito do RPi #3

```
twine==1.12.1
nose==1.3.7
pytest==3.8.2
pytest-runner==4.2
paho-mqtt>=1.4.0
hbmqtt== 0.9.5
scapy>=2.4.0
```

As dependências entre as classes da implementação estão apresentadas na Figura A.4:

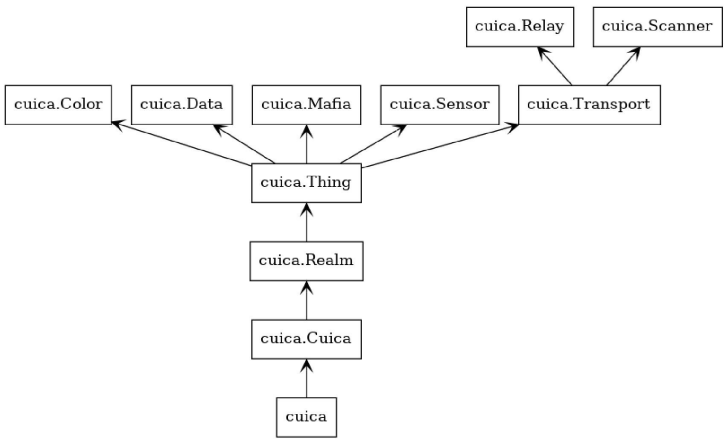


Figura A.4: Relação entre as classes.

O diagrama de classes estendido é apresentado na Figura A.5:

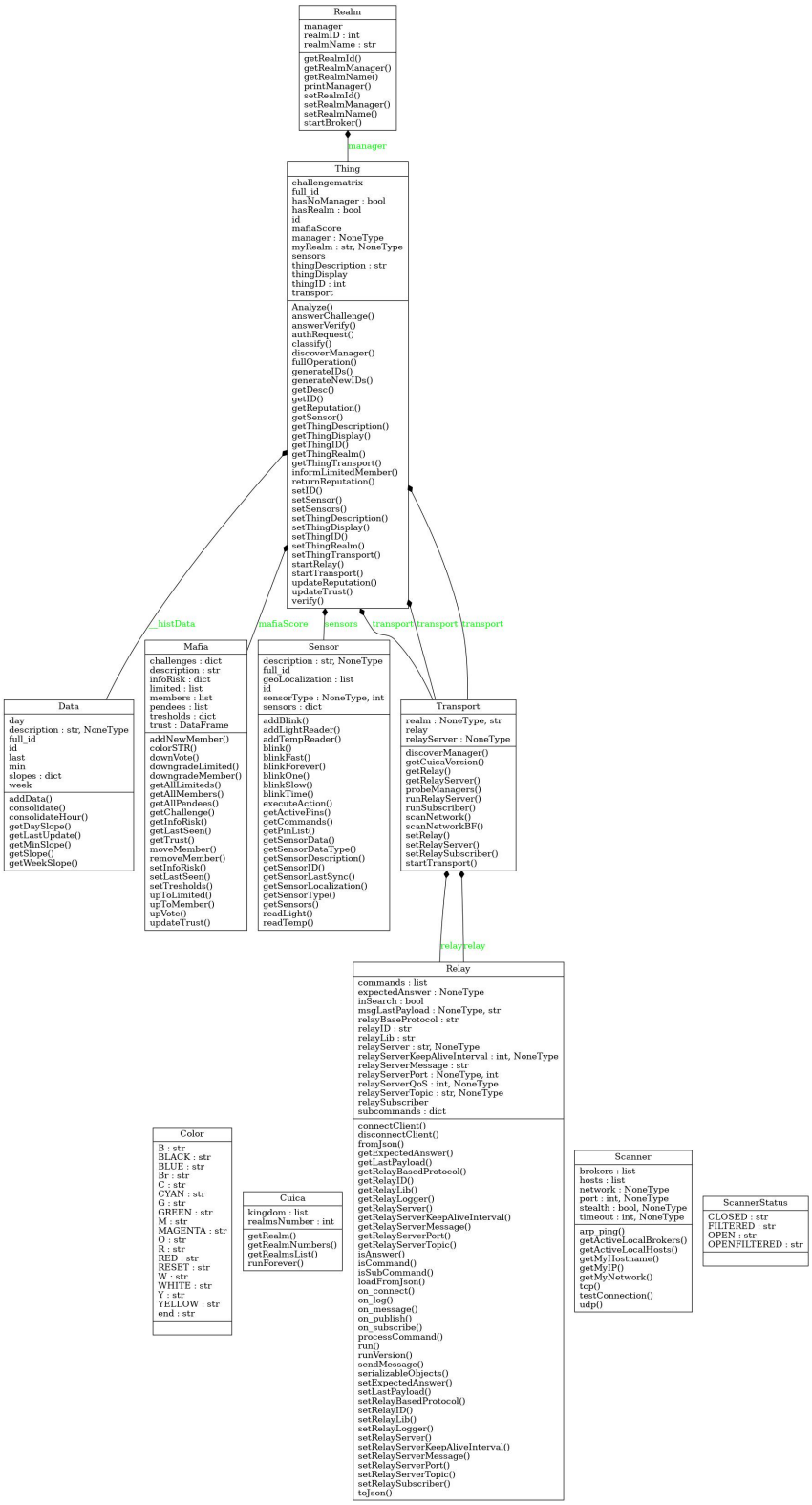


Figura A.5: Diagrama de classes.